P&S Heterogeneous Systems

Programming Heterogeneous Computing Systems with GPUs and other Accelerators

Dr. Juan Gómez Luna
Prof. Onur Mutlu
ETH Zürich

Fall 2022

3 October 2022

P&S: Heterogeneous Systems (I)

227-0085-51L Projects & Seminars: Programming Heterogeneous Computing Systems with GPUs and other Accelerators

Semester	Autumn Semester 2022
Lecturers	**O. Mutlu**, J. Gómez Luna
Periodicity	every semester recurring course
Language of instruction	English
Comment	Only for Electrical Engineering and Information Technology BSc. Course can only be registered for once. A repeatedly registration in a later semester is not chargeable.

Courses | Catalogue data | Performance assessment | Learning materials | Groups | Restrictions | Offered in | Overview

Abstract	The category of "Laboratory Courses, Projects, Seminars" includes courses and laboratories in various formats designed to impart practical knowledge and skills. Moreover, these classes encourage independent experimentation and design, allow for explorative learning and teach the methodology of project work.
Objective	The increasing difficulty of scaling the performance and efficiency of CPUs every year has created the need for turning computers into heterogeneous systems, i.e., systems composed of multiple types of processors that can suit better different types of workloads or parts of them. More than a decade ago, Graphics Processing Units (GPUs) became general-purpose parallel processors, in order to make their outstanding processing capabilities available to many workloads beyond graphics. GPUs have been a critical key to the recent rise of Machine Learning and Artificial Intelligence, which took unrealistic training times before the use of GPUs. Field-Programmable Gate Arrays (FPGAs) are another example computing device that can deliver impressive benefits in terms of performance and energy efficiency. More specific examples are (1) a plethora of specialized accelerators (e.g., Tensor Processing Units for neural networks), and (2) near-data processing architectures (i.e., placing compute capabilities near or inside memory/storage). Despite the great advances in the adoption of heterogeneous systems in recent years, there are still many challenges to tackle, for example: - Heterogeneous implementations (using GPUs, FPGAs, TPUs) of modern applications from important fields such as bioinformatics, machine learning, graph processing, medical imaging, personalized medicine, robotics, virtual reality, etc. - Scheduling techniques for heterogeneous systems with different general-purpose processors and accelerators, e.g., kernel offloading, memory scheduling, etc. - Workload characterization and programming tools that enable easier and more efficient use of heterogeneous systems. If you are enthusiastic about working hands-on with different software, hardware, and architecture projects for heterogeneous systems, this is your P&S. You will have the opportunity to program heterogeneous systems with different types of devices (CPUs, GPUs, FPGAs, TPUs), propose algorithmic changes to important applications to better leverage the compute power of heterogeneous systems, understand different workloads and identify the most suitable device for their execution, design optimized scheduling techniques, etc. In general, the goal will be to reach the highest performance reported for a given important application. The course is conducted in English. The course has two main parts: Weekly lectures on GPU and heterogeneous programming. Hands-on project: Each student develops his/her own project. Course website: **https://safari.ethz.ch/projects_and_seminars/doku.php?id=heterogeneous_systems** →

http://www.vvz.ethz.ch/Vorlesungsverzeichnis/lerneinheit.view?semkez=2022W&ansicht=KATALOGDATEN&lerneinheitId=164141&lang=en

P&S: Heterogeneous Systems (II)

The increasing difficulty of scaling the performance and efficiency of CPUs every year has created the need for turning computers into heterogeneous systems, i.e., systems composed of multiple types of processors that can suit better different types of workloads or parts of them. More than a decade ago, Graphics Processing Units (GPUs) became general-purpose parallel processors, in order to make their outstanding processing capabilities available to many workloads beyond graphics. GPUs have been critical key to the recent rise of Machine Learning and Artificial Intelligence, which took unrealistic training times before the use of GPUs. Field-Programmable Gate Arrays (FPGAs) are another example computing device that can deliver impressive benefits in terms of performance and energy efficiency. More specific examples are (1) a plethora of specialized accelerators (e.g., Tensor Processing Units for neural networks), and (2) near-data processing architectures (i.e., placing compute capabilities near or inside memory/storage).

Despite the great advances in the adoption of heterogeneous systems in recent years, there are still many challenges to tackle, for example:

- Heterogeneous implementations (using GPUs, FPGAs, TPUs) of modern applications from important fields such as bioinformatics, machine learning, graph processing, medical imaging, personalized medicine, robotics, virtual reality, etc.
- Scheduling techniques for heterogeneous systems with different general-purpose processors and accelerators, e.g., kernel offloading, memory scheduling, etc.
- Workload characterization and programming tools that enable easier and more efficient use of heterogeneous systems.

Flynn's Taxonomy of Computers

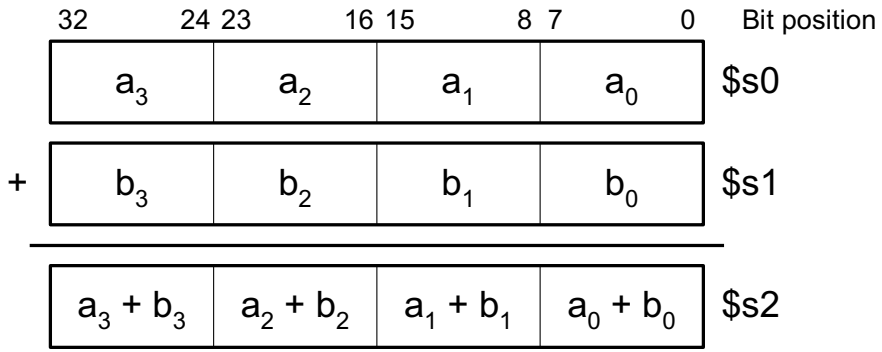
- Mike Flynn, "Very High-Speed Computing Systems," Proc. of IEEE, 1966

- SISD: Single instruction operates on single data element
- SIMD: Single instruction operates on multiple data elements
  - Array processor
  - Vector processor
- MISD: Multiple instructions operate on single data element
  - Closest form: systolic array processor, streaming processor
- MIMD: Multiple instructions operate on multiple data elements (multiple instruction streams)
  - Multiprocessor
  - Multithreaded processor

SIMD ISA Extensions

- Single Instruction Multiple Data (SIMD) extension instructions
  - Single instruction acts on multiple pieces of data at once
  - Common application: graphics
  - Perform short arithmetic operations (also called *packed arithmetic*)
- For example: add four 8-bit numbers
- Must modify ALU to eliminate carries between 8-bit values

```
padd8 $s2, $s0, $s1
```



Intel Pentium MMX Operations

- Idea: One instruction operates on multiple data elements simultaneously
  - *À la* array processing (yet much more limited)
  - Designed with multimedia (graphics) operations in mind

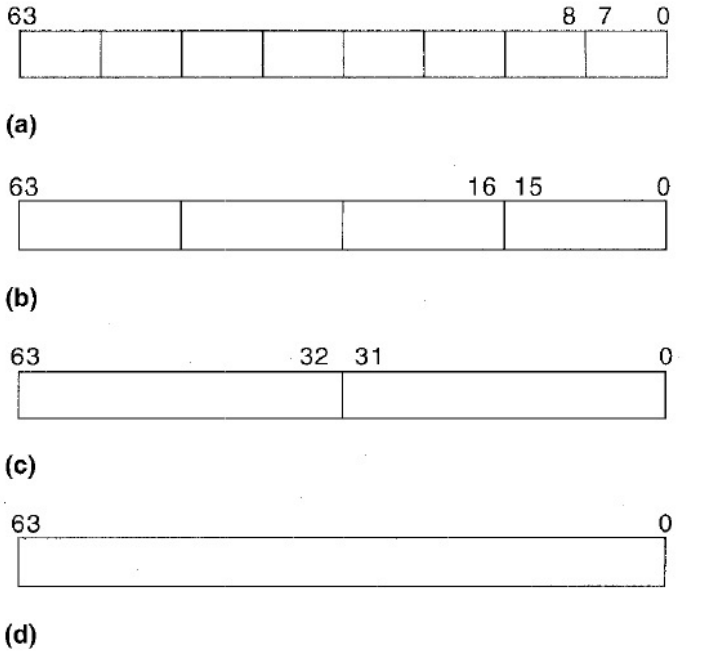


Figure 1. MMX technology data types: packed byte (a), packed word (b), packed doubleword (c), and quadword (d).

No VLEN register
Opcode determines data type:
8 8-bit bytes
4 16-bit words
2 32-bit doublewords
1 64-bit quadword

Stride is always equal to 1.

Peleg and Weiser, "MMX Technology Extension to the Intel Architecture," IEEE Micro, 1996.

MMX Example: Image Overlaying (I)

- Goal: Overlay the human in image x on top of the background in image y

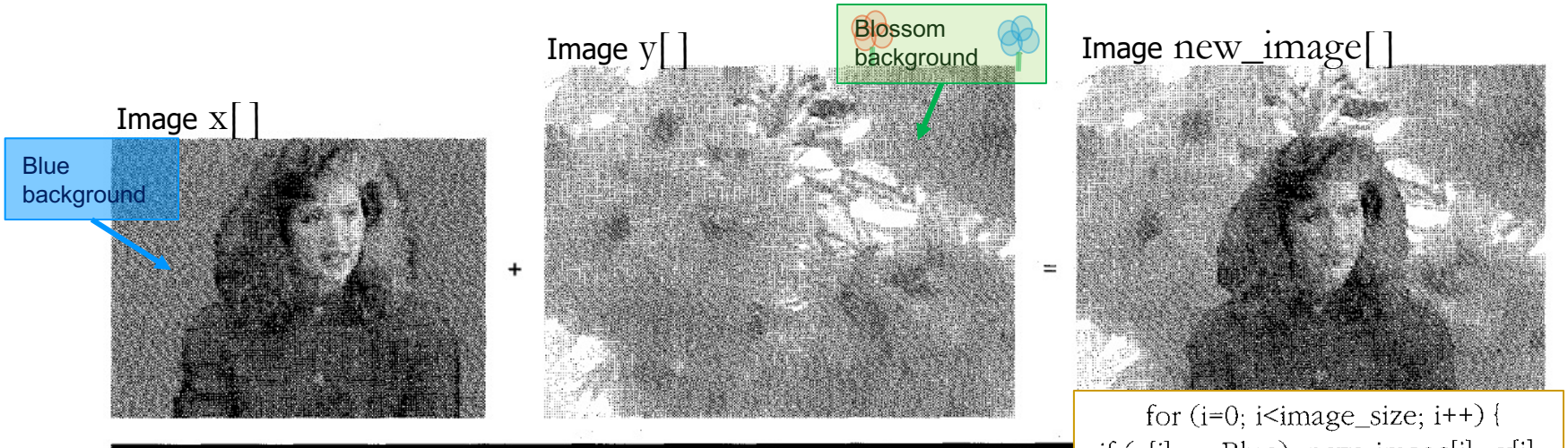


Figure 8. Chroma keying: image overlay using a background color.

```
for (i=0; i<image_size; i++) {
if (x[i] == Blue)  new_image[i] =y[i];
        else new_image[i] = x[i];
}
```

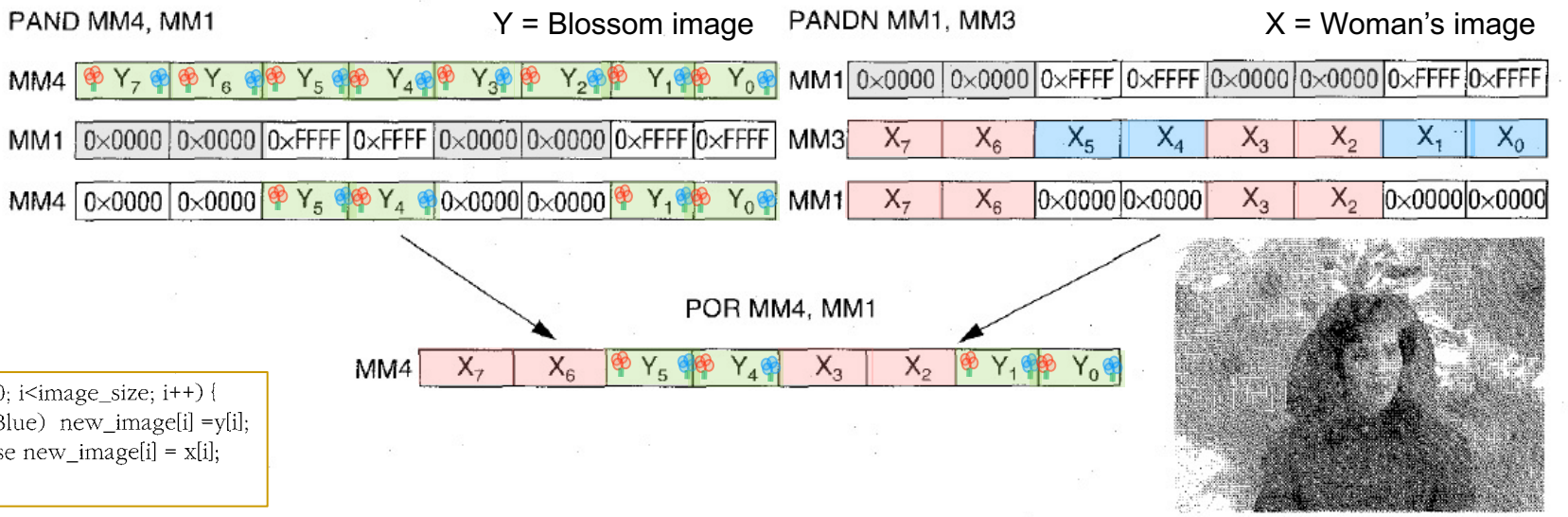
PCMPEQB MM1, MM3

MM1	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue
Image x[ ] MM3	X7!=blue	X6!=blue	X5=blue	X4=blue	X3!=blue	X2!=blue	X1=blue	X0=blue
Bit mask MM1	0x0000	0x0000	0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF

Bitmask

Figure 9. Generating the selection bit mask.

MMX Example: Image Overlaying (II)




```
for (i=0; i<image_size; i++) {
if (x[i] == Blue)  new_image[i] =y[i];
        else new_image[i] = x[i];
}
```

Figure 10. Using the mask with logical MMX instructions to perform a conditional select.

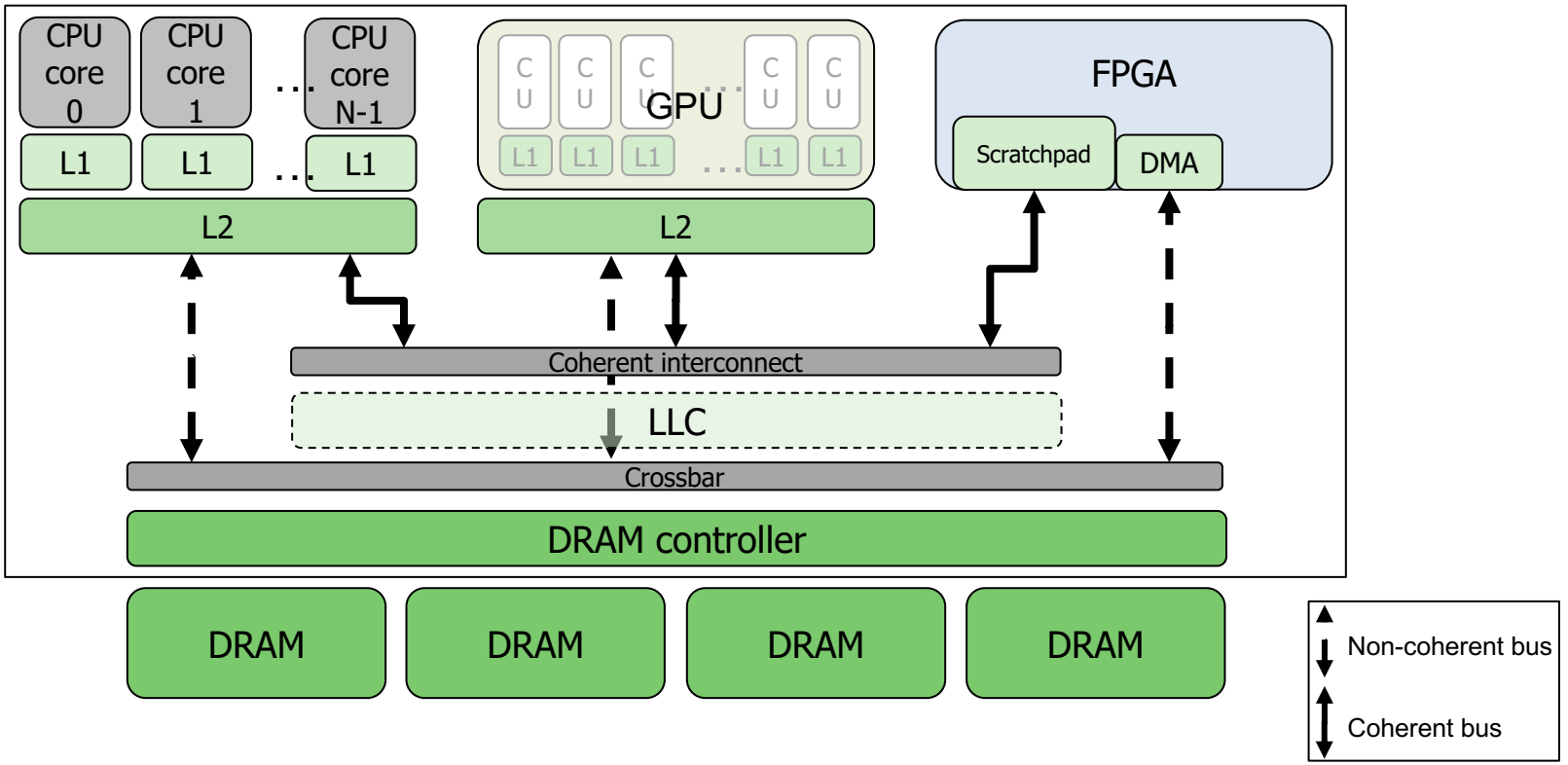
```
Movq    mm3, mem1    /* Load eight pixels from
                        woman's image
Movq    mm4, mem2    /* Load eight pixels from the
                        blossom image
Pcmpeqb mm1, mm3
Pand    mm4, mm1
Pandn   mm1, mm3
Por     mm4, mm1
```

Figure 11. MMX code sequence for performing a conditional select.

Peleg and Weiser, "MMX Technology Extension to the Intel Architecture," IEEE Micro, 1996.

Heterogeneous Computing Systems

- The end of Moore's law created the need for heterogeneous systems
  - More suitable devices for each type of workload
  - Increased performance and energy efficiency

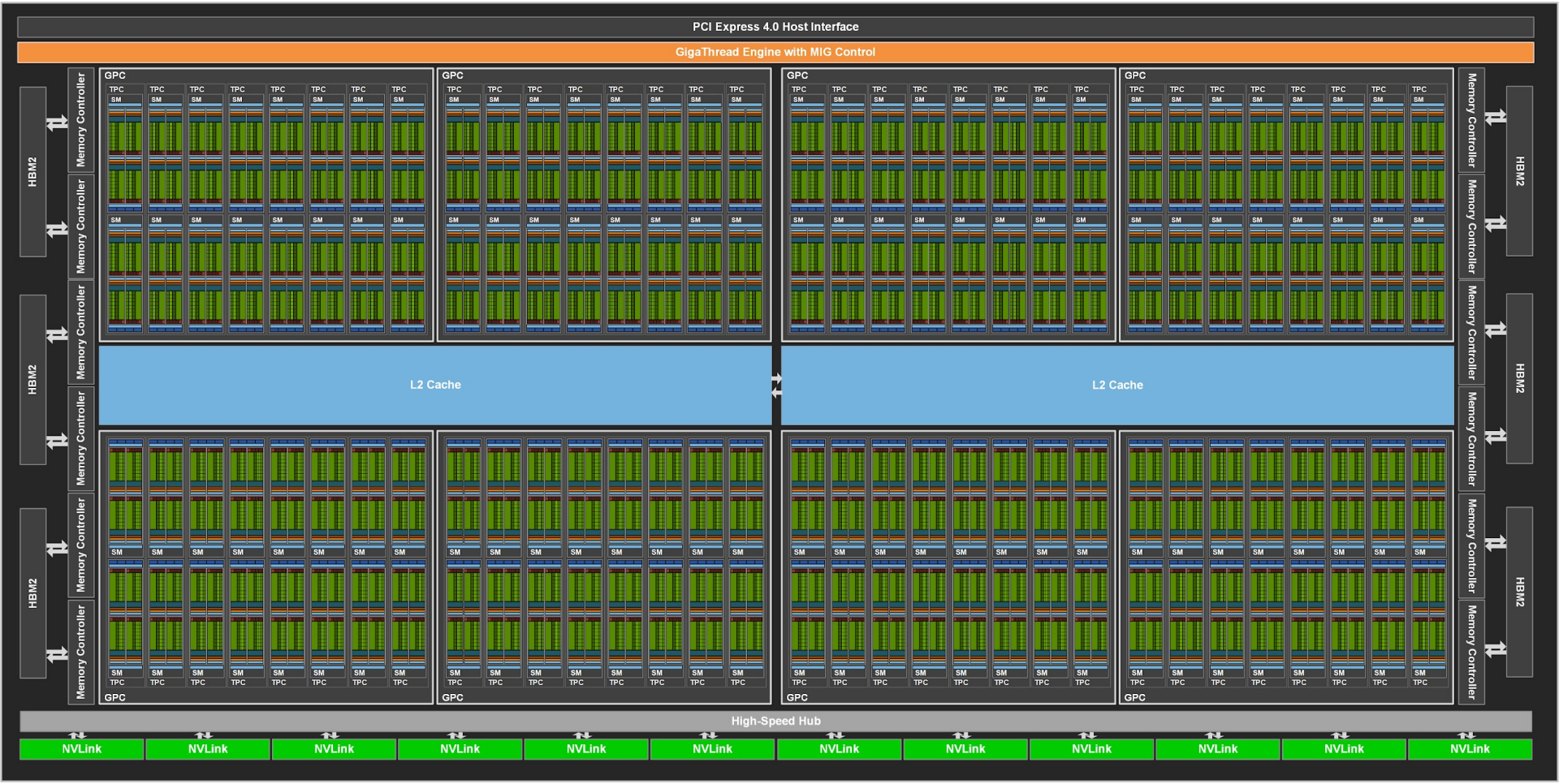


Goals of this P&S Course

P&S Heterogeneous Systems: Contents

- We will introduce the need for heterogeneity in current computing systems, in order to achieve high performance and energy efficiency
- You will get familiar with some of the different heterogeneous devices that are available in computing systems
- You will learn workload distribution and parallelization strategies that leverage heterogeneous devices
- You will work hands-on: analyzing workloads, programming heterogeneous architectures, proposing scheduling/offloading mechanisms, etc.

NVIDIA A100 (2020)



https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

108 cores on the A100

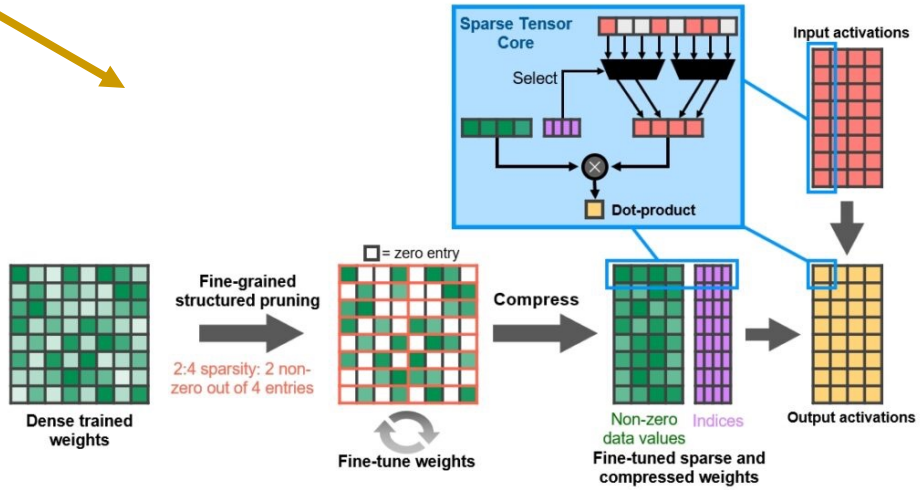
(Up to 128 cores in the full-blown chip)

40MB L2 cache

NVIDIA A100 Core

GPU compute throughput:

- 19.5 TFLOPS Single Precision
- 9.7 TFLOPS Double Precision
- 312 TFLOPS for Deep Learning (Tensor cores)



https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

NVIDIA H100 Block Diagram

https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/

144 cores on the full GH100

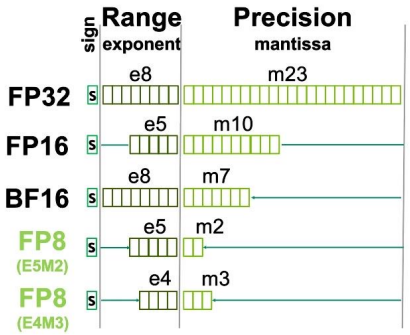
60MB L2 cache

NVIDIA H100 Core

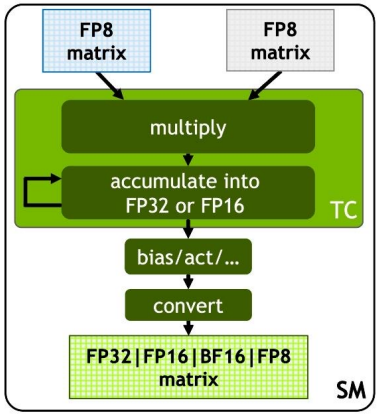
48 TFLOPS Single Precision*

24 TFLOPS Double Precision*

800 TFLOPS (FP16, Tensor Cores)*



Allocate 1 bit to either range or precision

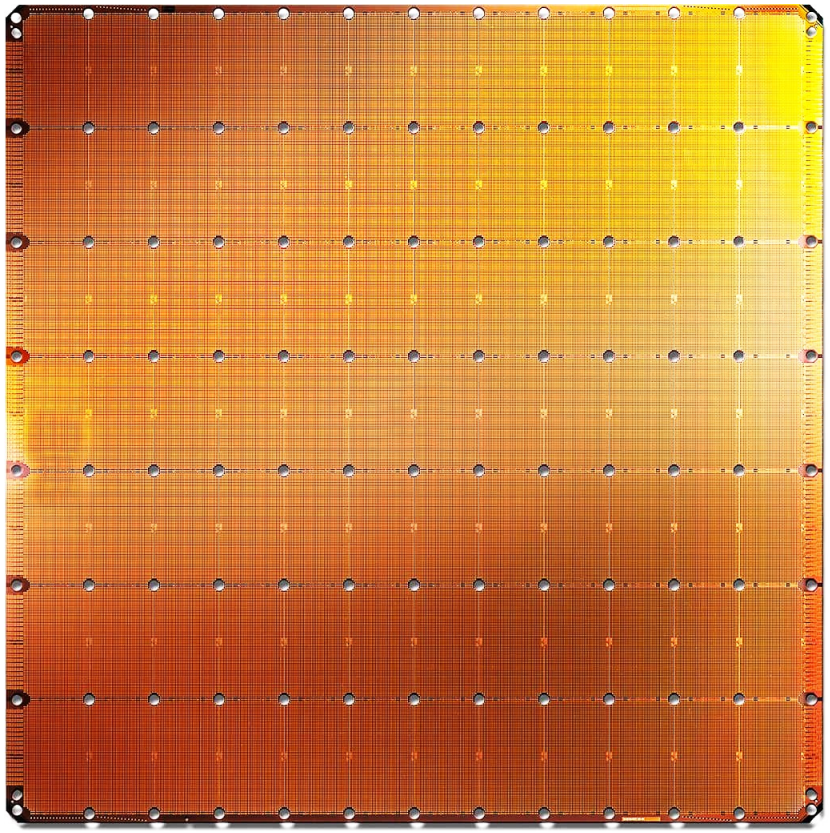


Support for multiple accumulator and output types

https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/

* Preliminary performance estimates

Cerebras's Wafer Scale Engine (2019)



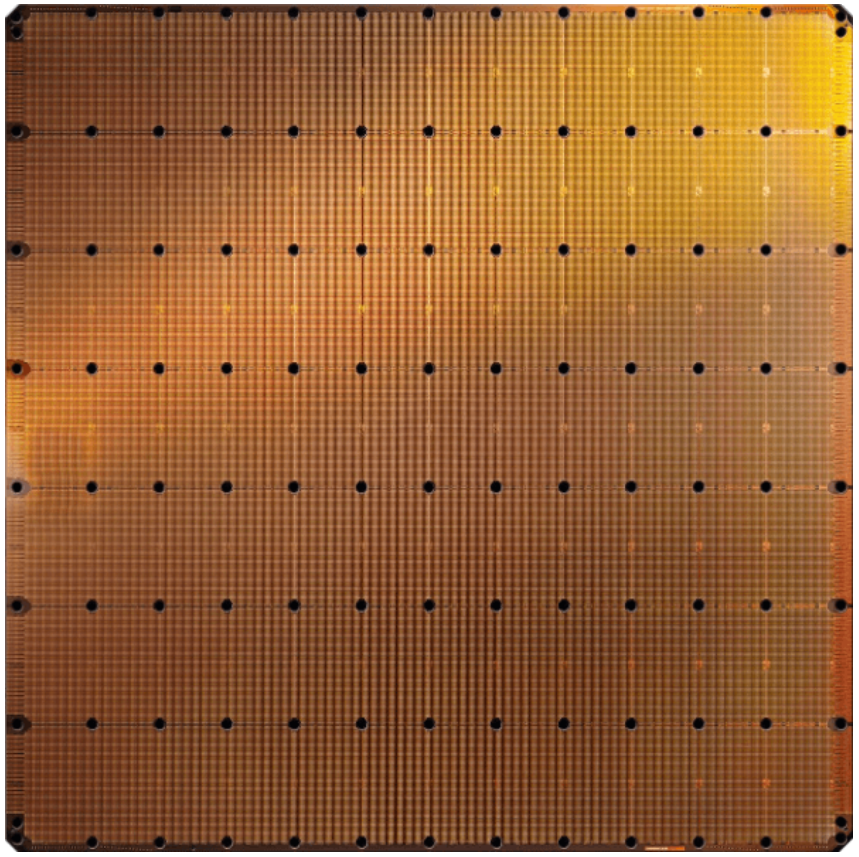
- The largest ML accelerator chip (2019)
- 400,000 cores

Cerebras WSE
1.2 Trillion transistors
46,225 mm^2

Largest GPU
21.1 Billion transistors
815 mm^2
NVIDIA TITAN V

https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning
https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/

Cerebras's Wafer Scale Engine-2 (2021)



- The largest ML accelerator chip (2021)
- 850,000 cores

Cerebras WSE-2
2.6 Trillion transistors
46,225 mm^2

Largest GPU
54.2 Billion transistors
826 mm^2
NVIDIA Ampere GA100

https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning
https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/

SAFARI Live Seminar: Sean Lie

SAFARI Live Seminar: Sean Lie, 28 Feb 2022

Posted on January 19, 2022 by ewent

Join us for our SAFARI Live Seminar with Sean Lie, Cerebras Systems
Monday, February 28 2022 at 6:00 pm Zurich time (CET)

Sean Lie, co-founder and Chief Hardware Architect at Cerebras Systems
Thinking Outside the Die: Architecting the ML Accelerator of the Future

Livestream on YouTube Link

https://safari.ethz.ch/safari-live-seminar-sean-lie-28-feb-2022/

Google TPU Generation I (~2016)

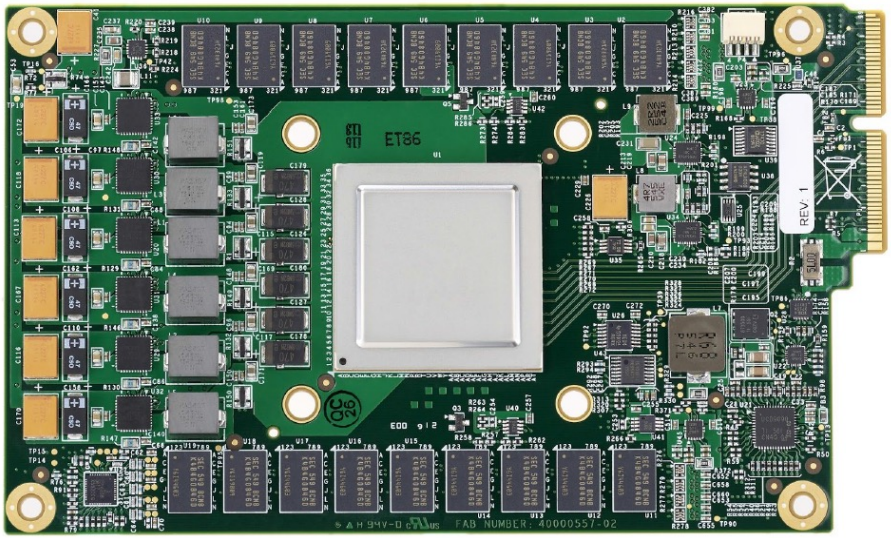


Figure 3. TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

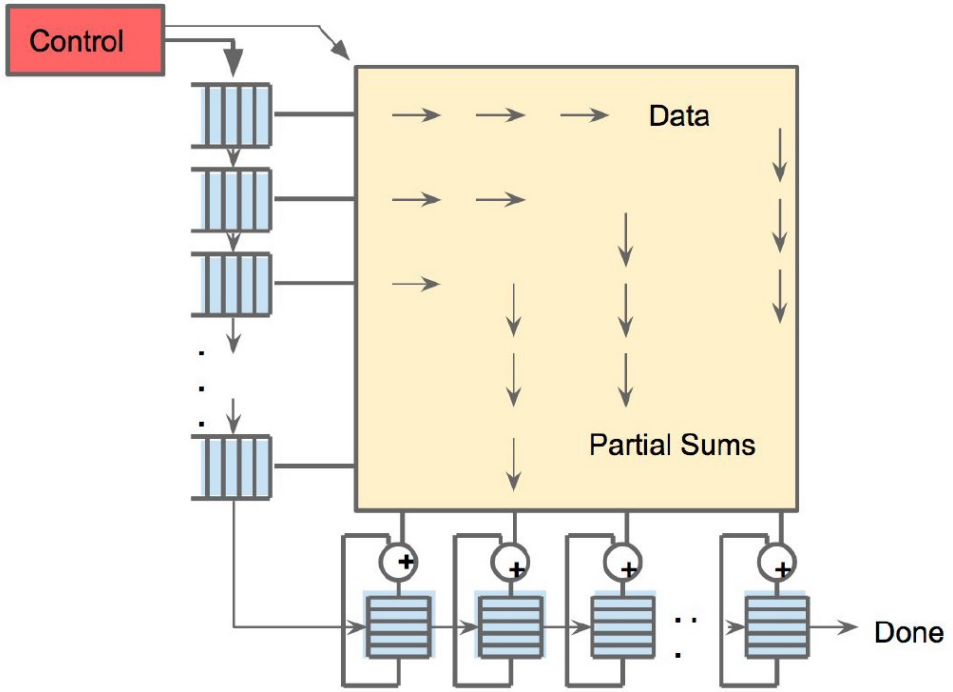
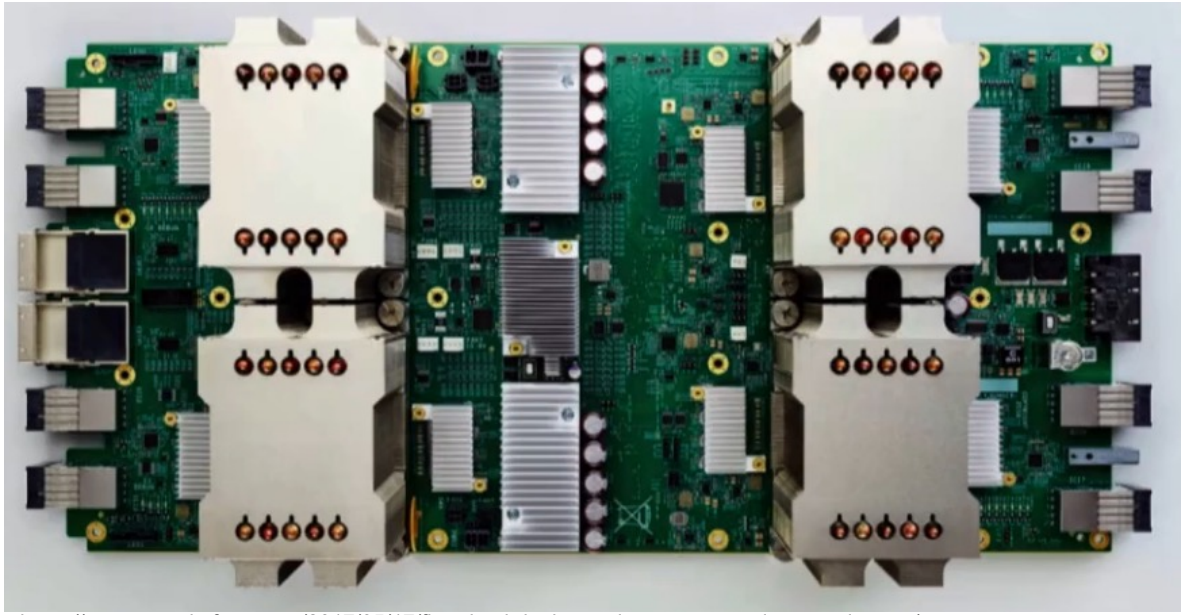


Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", ISCA 2017.

Google TPU Generation II (2017)



https://www.nextplatform.com/2017/05/17/first-depth-look-googles-new-second-generation-tpu/

4 TPU chips
vs 1 chip in TPU1

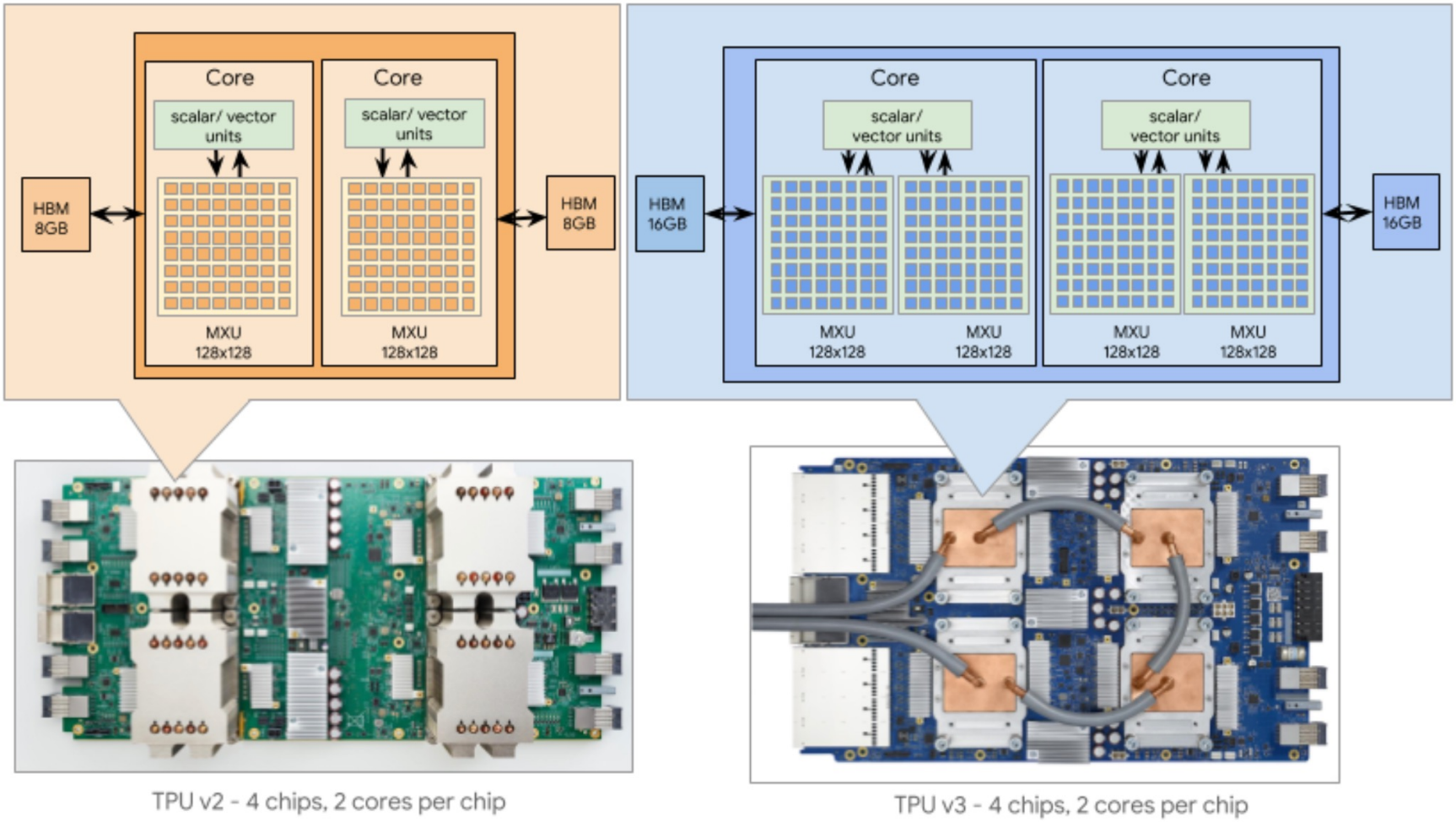
High Bandwidth Memory
vs DDR3

Floating point operations
vs FP16

45 TFLOPS per chip
vs 23 TOPS

Designed for training
and inference
vs only inference

Google TPU Generation III (2019)



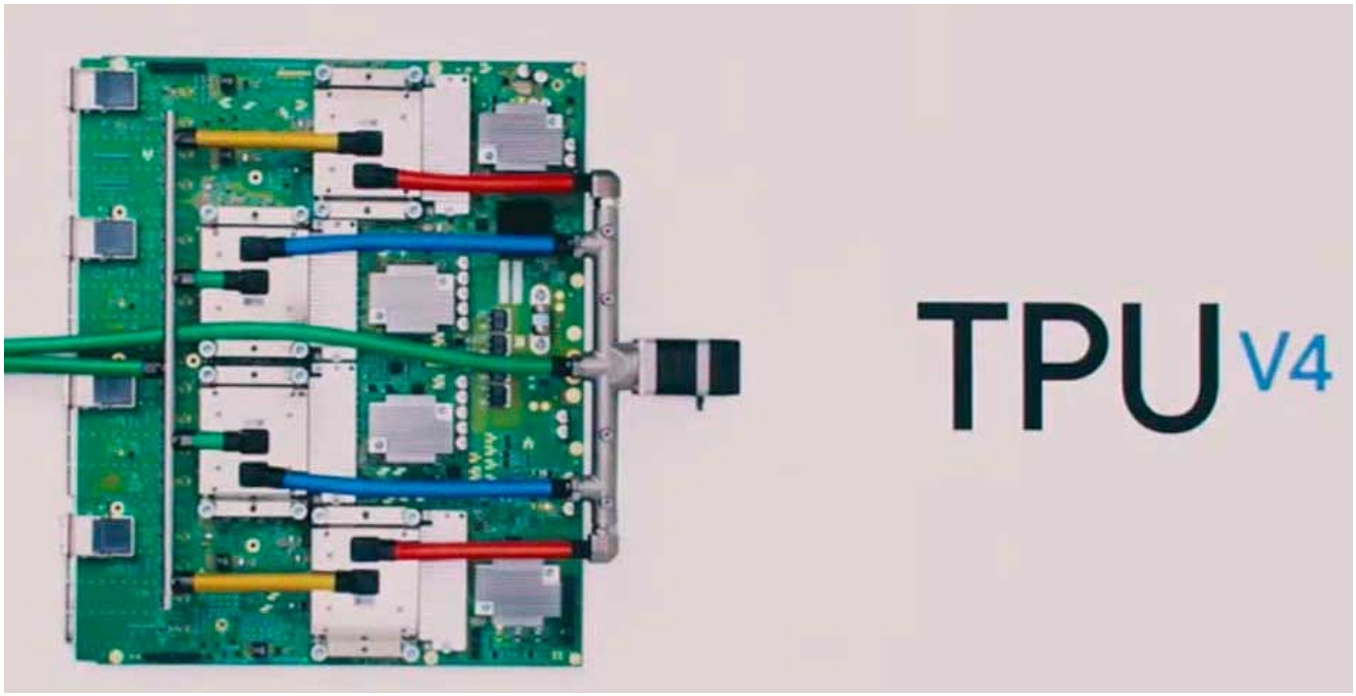
32GB HBM per chip
vs 16GB HBM in TPU2

4 Matrix Units per chip
vs 2 Matrix Units in TPU2

90 TFLOPS per chip
vs 45 TFLOPS in TPU2

https://cloud.google.com/tpu/docs/system-architecture

Google TPU Generation IV (2019)



New ML applications (vs. TPU3):

- Computer vision
- Natural Language Processing (NLP)
- Recommender system
- Reinforcement learning that plays Go

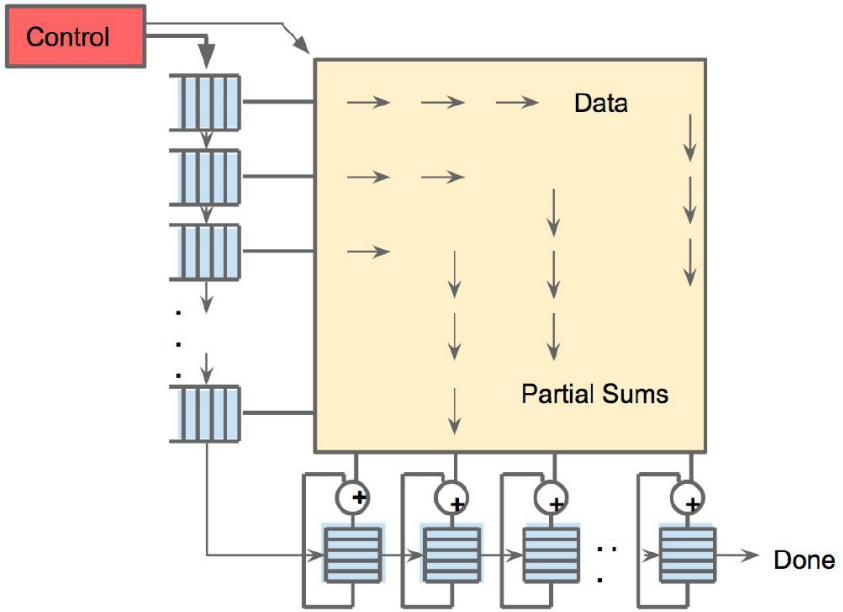
250 TFLOPS per chip in 2021
vs 90 TFLOPS in TPU3

1 ExaFLOPS per board

https://spectrum.ieee.org/tech-talk/computing/hardware/heres-how-googles-tpu-v4-ai-chip-stacked-up-in-training-tests

An Example Modern Systolic Array: TPU (II)

As reading a large SRAM uses much more power than arithmetic, the matrix unit uses systolic execution to save energy by reducing reads and writes of the Unified Buffer [Kun80][Ram91][Ovt15b]. Figure 4 shows that data flows in from the left, and the weights are loaded from the top. A given 256-element multiply-accumulate operation moves through the matrix as a diagonal wavefront. The weights are preloaded, and take effect with the advancing wave alongside the first data of a new block. Control and data are pipelined to give the illusion that the 256 inputs are read at once, and that they instantly update one location of each of 256 accumulators. From a correctness perspective, software is unaware of the systolic nature of the matrix unit, but for performance, it does worry about the latency of the unit.



Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", ISCA 2017.

An Example Modern Systolic Array: TPU (III)

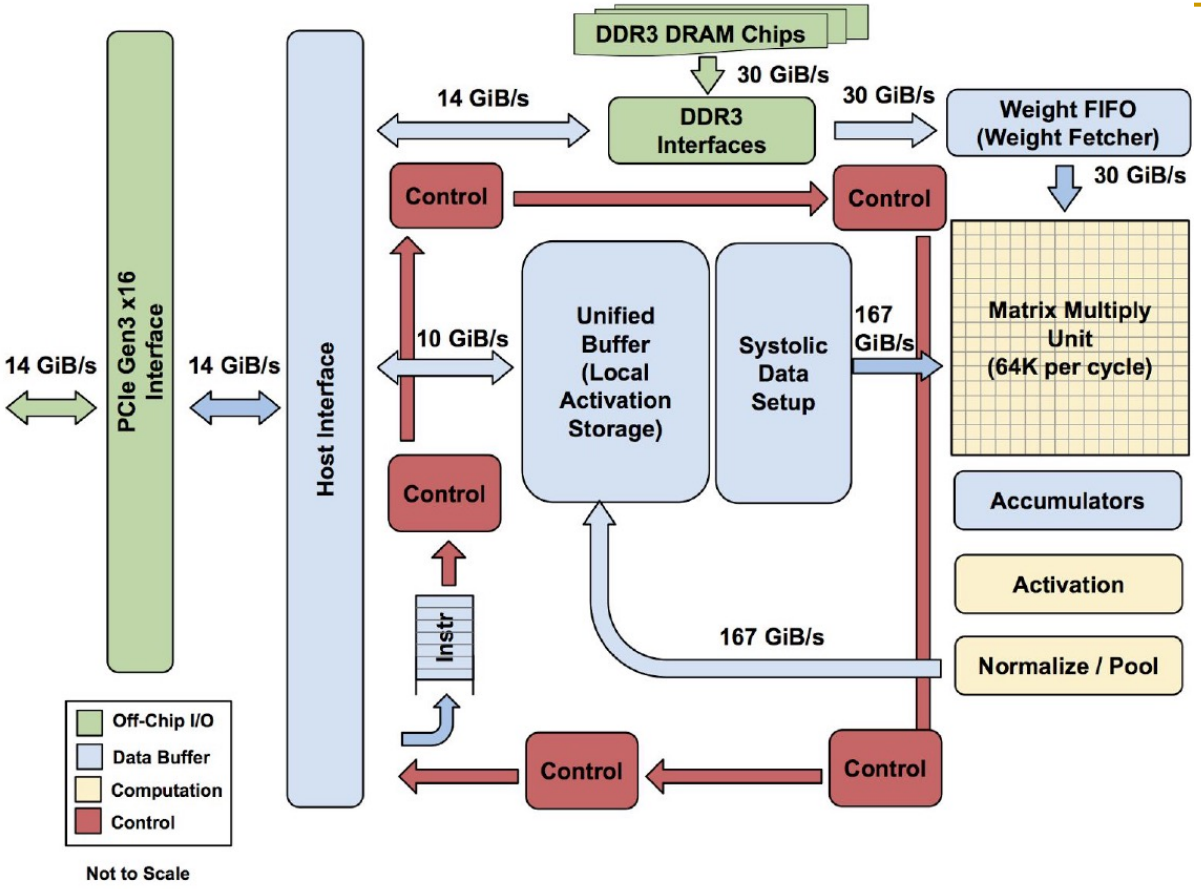
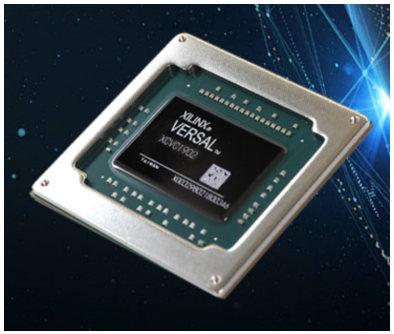
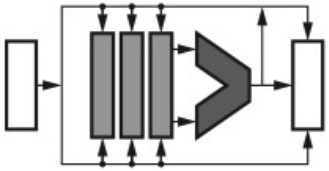
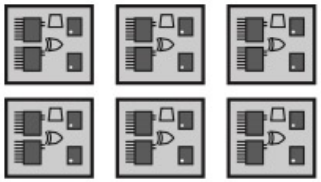
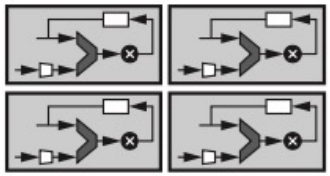


Figure 1. TPU Block Diagram. The main computation part is the yellow Matrix Multiply unit in the upper right hand corner. Its inputs are the blue Weight FIFO and the blue Unified Buffer (UB) and its output is the blue Accumulators (Acc). The yellow Activation Unit performs the nonlinear functions on the Acc, which go to the UB.

Xilinx Versal ACAP (2020) (I)

- Three compute engines inside the same chip
  - Different workloads, different devices

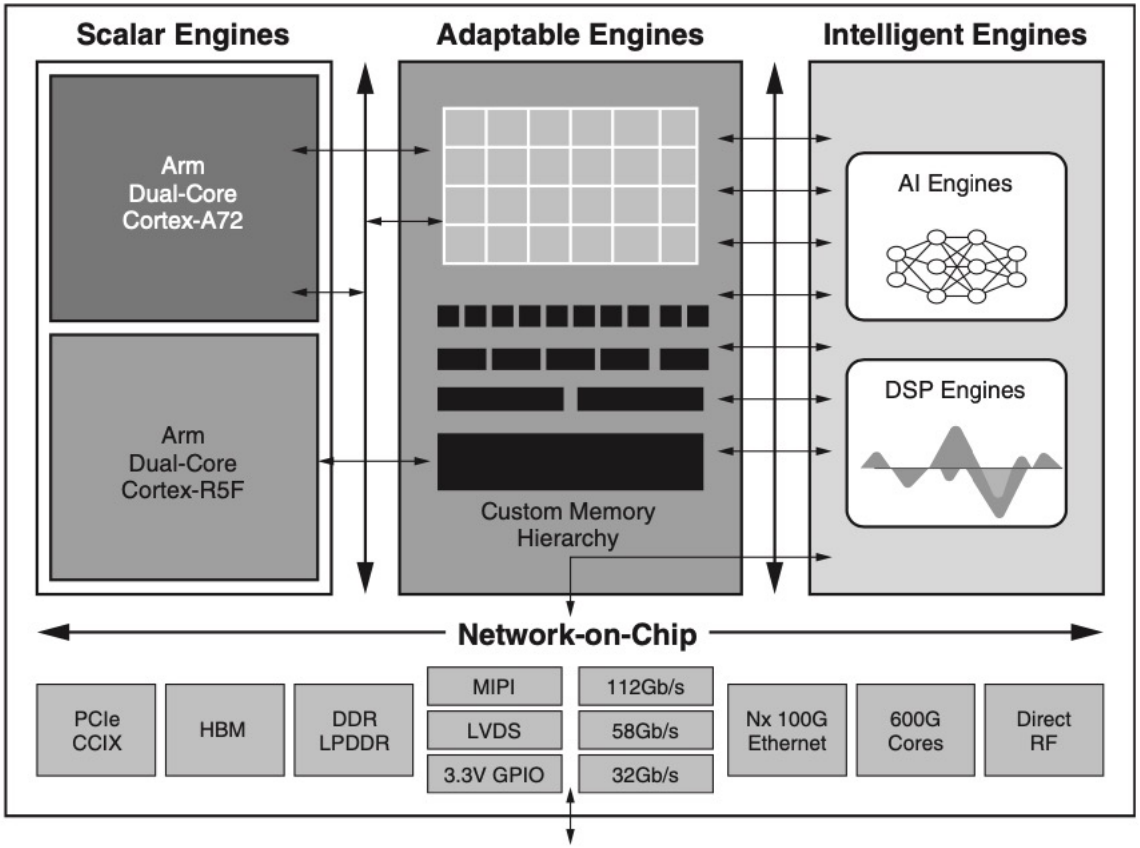


Scalar Processing	Adaptable Hardware	Vector Processing (e.g., GPU, DSP)
		
Complex Algorithms and Decision Making	Processing of Irregular Data Structures *Genomic Sequencing*	Domain-specific Parallelism
	Latency Critical Workloads *Real-Time Control*	Signal Processing *Complex Math, Convolutions*
	Sensor Fusion *Pre-processing, Programmable I/O*	Video and Image Processing

https://www.xilinx.com/products/silicon-devices/acap/versal.html

Xilinx Versal ACAP (2020) (II)

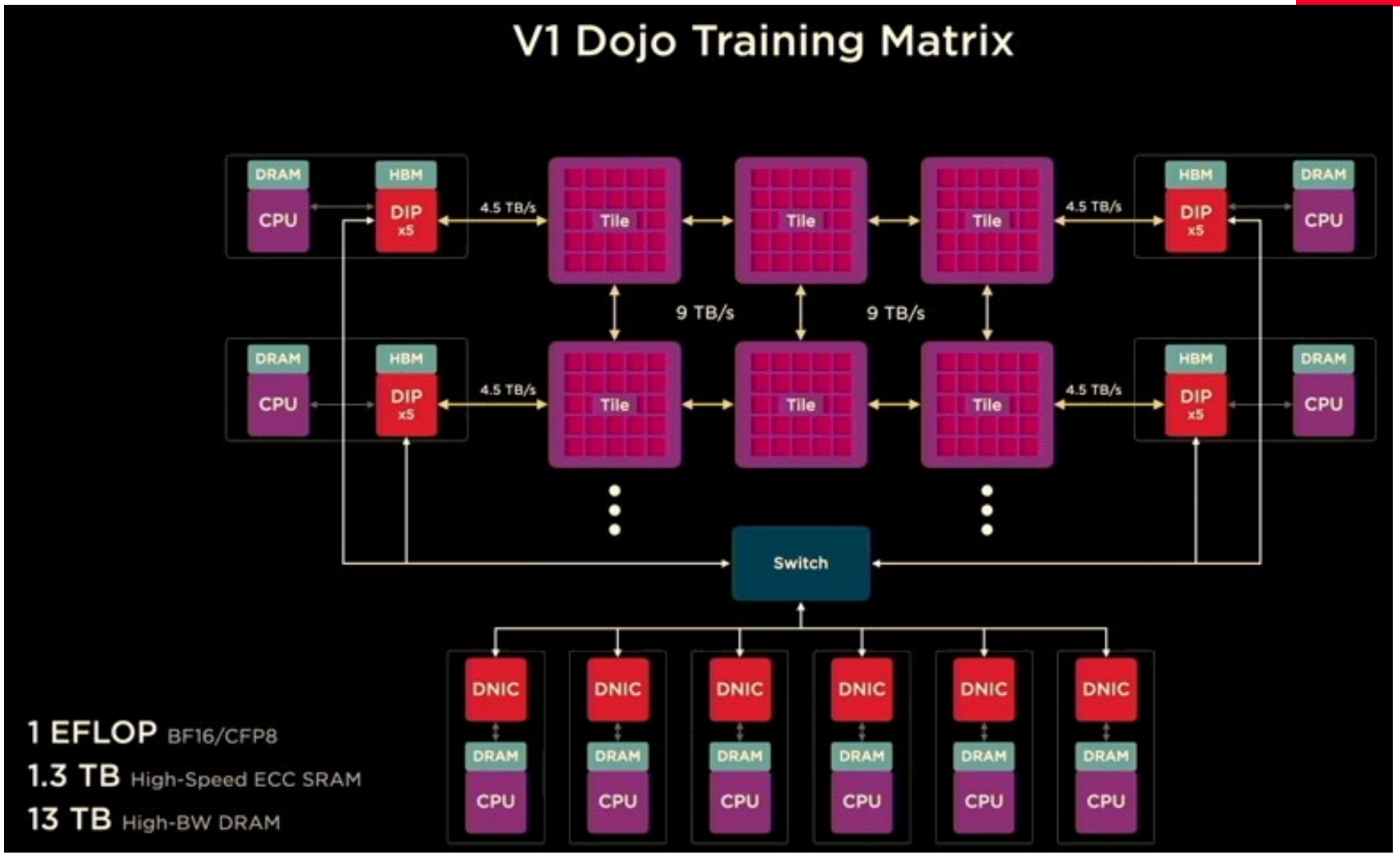
- Three compute engines inside the same chip
  - Scalar cores, reconfigurable engines, vector processors



https://www.xilinx.com/products/silicon-devices/acap/versal.html

Tesla Dojo (2022) (I)

- Tesla Dojo Chip & System



https://www.nextplatform.com/2022/08/23/inside-teslas-innovative-and-homegrown-dojo-ai-supercomputer/

Tesla Dojo (2022) (II)

- Tesla Dojo Chip & System

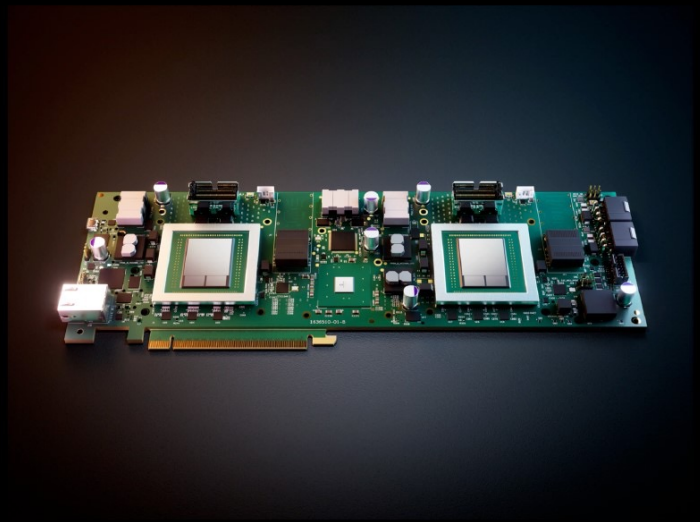
V1 Dojo Interface Processor

32GB High-Bandwidth Memory
- 800 GB/s Total Memory Bandwidth

900 GB/s TTP Interface
- Tesla Transport Protocol (TTP) - Full custom protocol
- Provides full DRAM bandwidth to Training Tile

50 GB/s TTP over Ethernet (TTPoE)
- Enables extending communication over standard Ethernet
- Native hardware support

32 GB/s Gen4 PCIe Interface

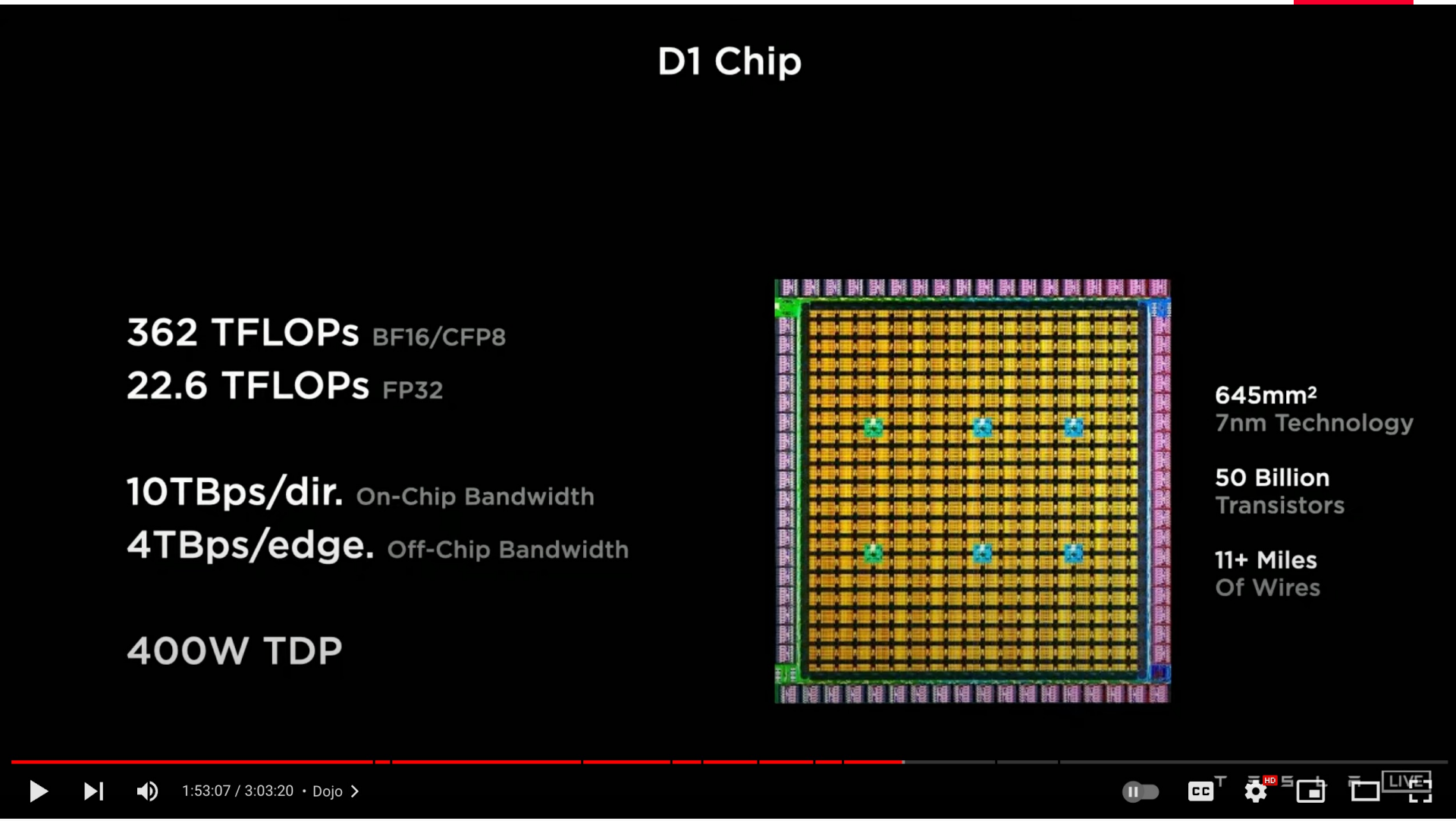


TESLA

https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s

Tesla Dojo (2022) (III)

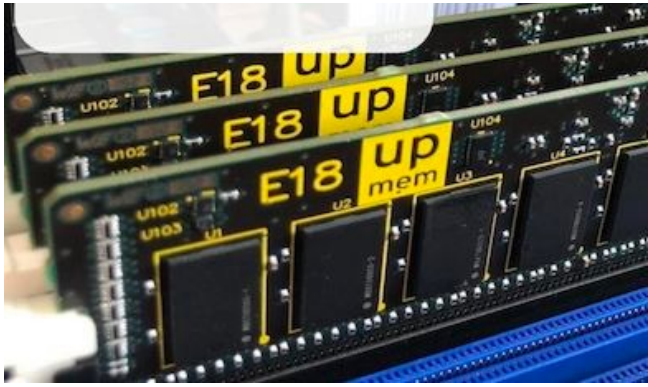
- Tesla Dojo Chip & System



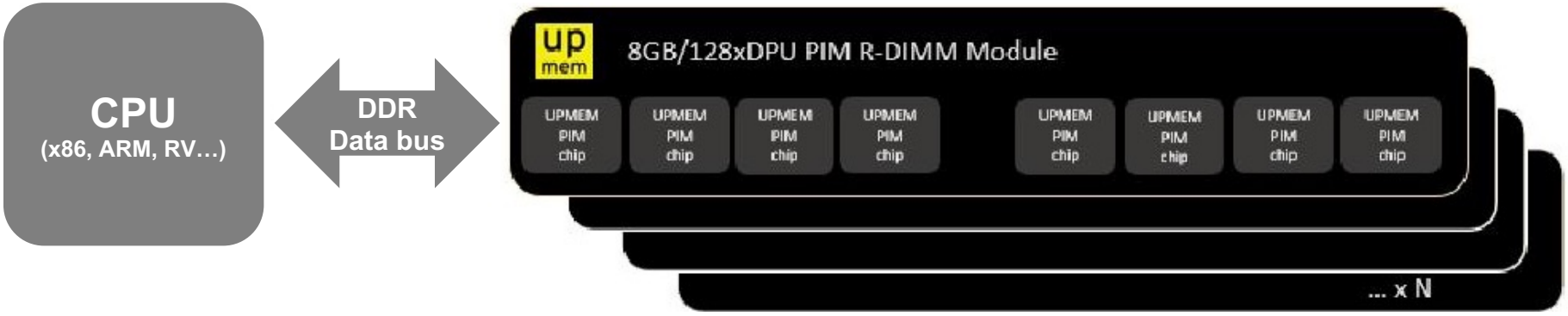
https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s

UPMEM Processing-in-DRAM Engine (2019)

- Processing in DRAM Engine
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.



- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth



https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem
https://www.upmem.com/video-upmem-presenting-its-true-processing-in-memory-solution-hot-chips-2019/

Samsung Function-in-Memory DRAM (2021)

Samsung Newsroom

CORPORATE | PRODUCTS | PRESS RESOURCES | VIEWS | ABOUT US

Samsung Develops Industry's First High Bandwidth Memory with AI Processing Power

Korea on February 17, 2021 Audio Share

The new architecture will deliver over twice the system performance and reduce energy consumption by more than 70%

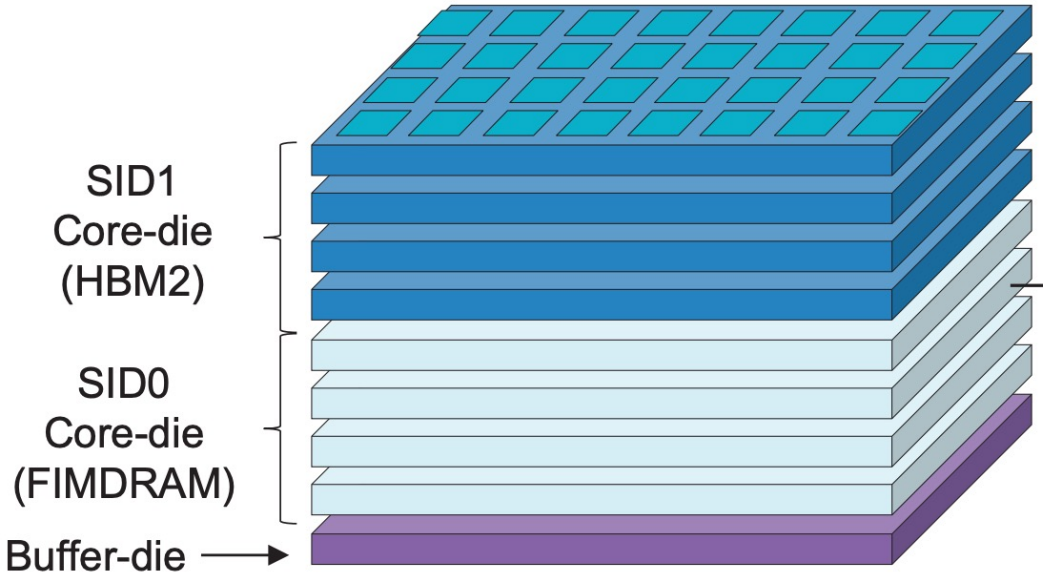
Samsung Electronics, the world leader in advanced memory technology, today announced that it has developed the industry's first High Bandwidth Memory (HBM) integrated with artificial intelligence (AI) processing power — the HBM-PIM. The new processing-in-memory (PIM) architecture brings powerful AI computing capabilities inside high-performance memory, to accelerate large-scale processing in data centers, high performance computing (HPC) systems and AI-enabled mobile applications.

Kwangil Park, senior vice president of Memory Product Planning at Samsung Electronics stated, "Our groundbreaking HBM-PIM is the industry's first programmable PIM solution tailored for diverse AI-driven workloads such as HPC, training and inference. We plan to build upon this breakthrough by further collaborating with AI solution providers for even more advanced PIM-powered applications."

https://news.samsung.com/global/samsung-develops-industrys-first-high-bandwidth-memory-with-ai-processing-power

Samsung Function-in-Memory DRAM (2021)

- FIMDRAM based on HBM2



[3D Chip Structure of HBM with FIMDRAM]

Chip Specification
128DQ / 8CH / 16 banks / BL4
32 PCU blocks (1 FIM block/2 banks)
1.2 TFLOPS (4H)
FP16 ADD / Multiply (MUL) / Multiply-Accumulate (MAC) / Multiply-and- Add (MAD)

ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

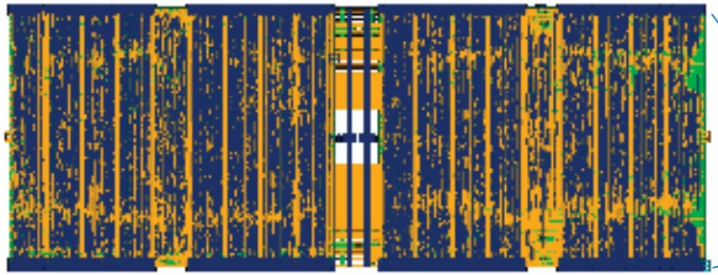
Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1], David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3], Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung Electronics, Suwon, Korea

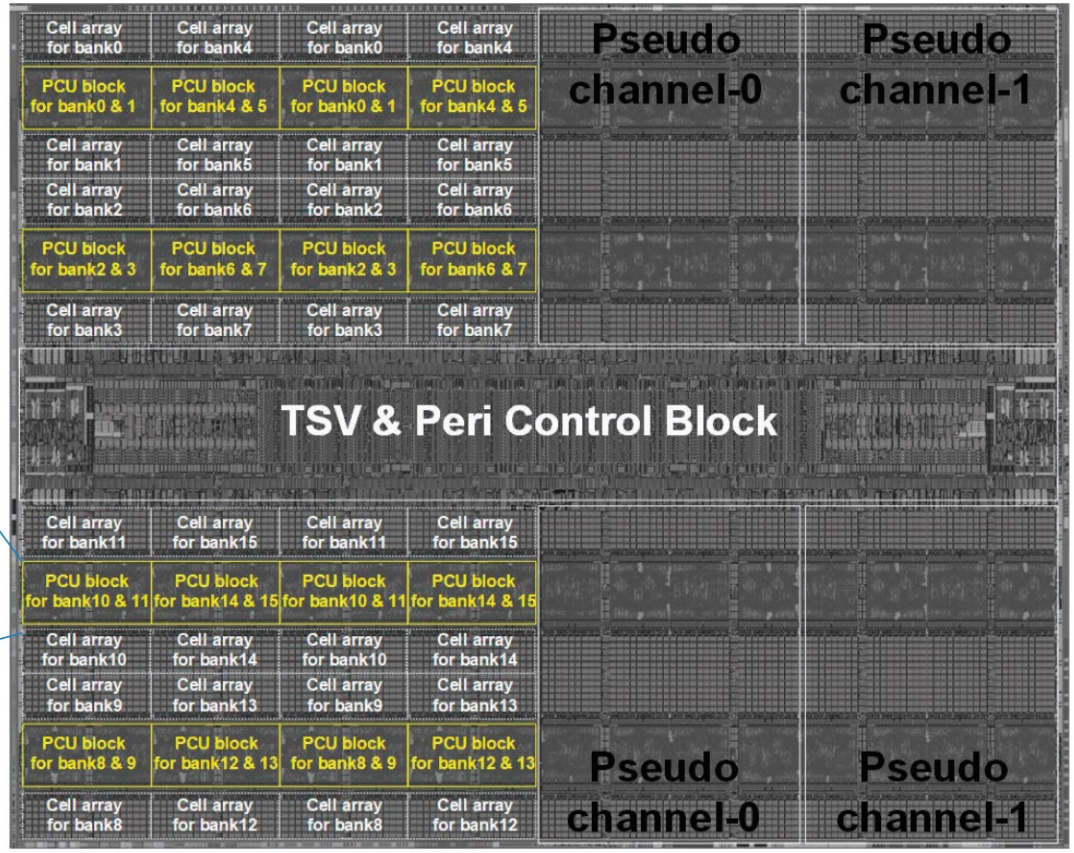
Samsung Function-in-Memory DRAM (2021)

Chip Implementation

- Mixed design methodology to implement FIMDRAM
  - Full-custom + Digital RTL



[Digital RTL design for PCU block]



ISSCC 2021 / SESSION 25 / DRAM / 25.4

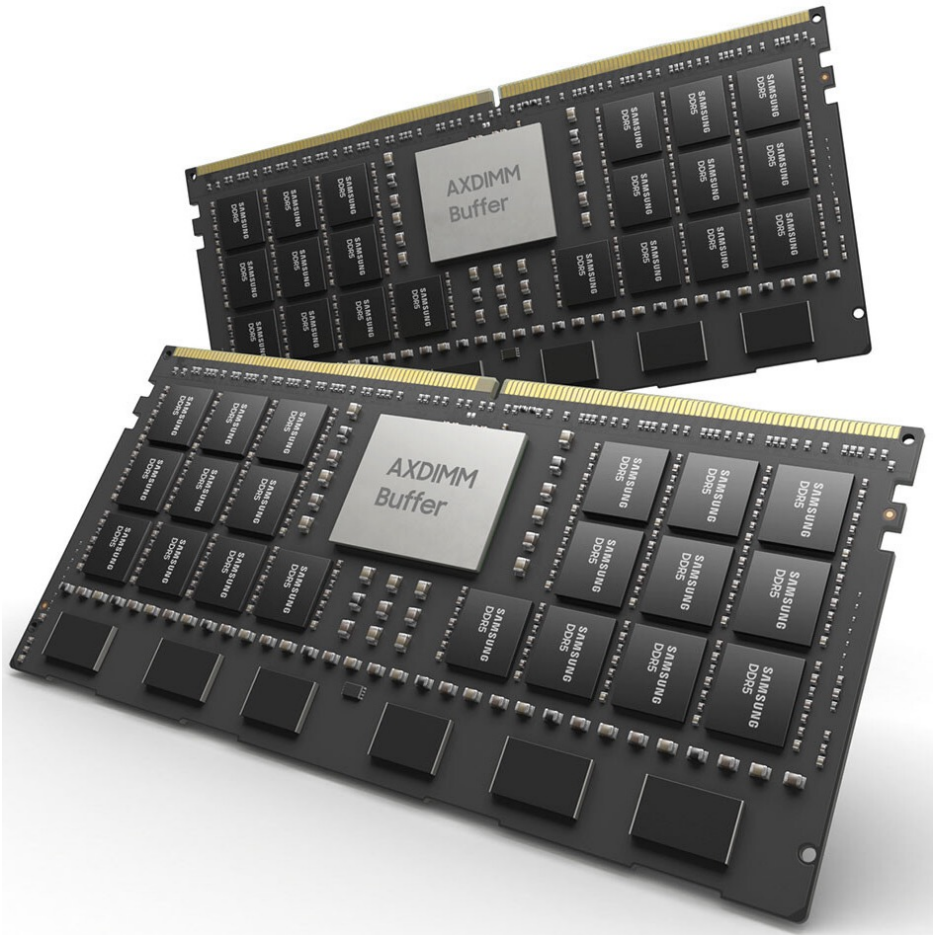
25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

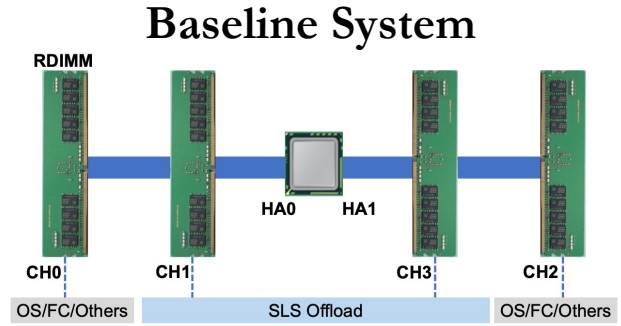
Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1], David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3], Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

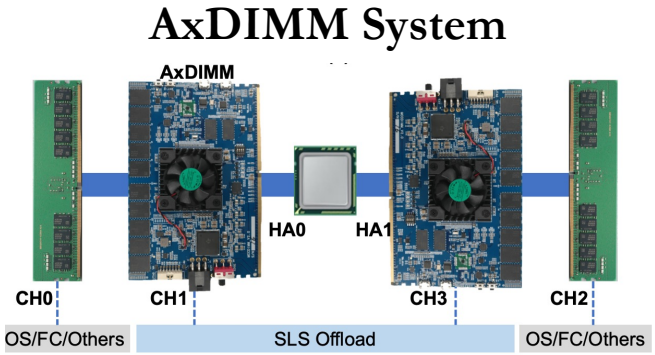
[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung Electronics, Suwon, Korea

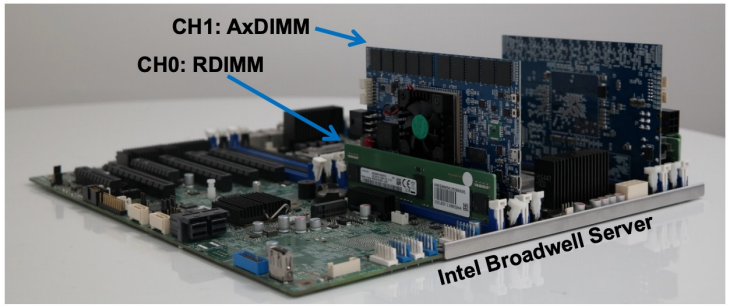
Samsung AxDIMM (2021)

- DIMM-based PIM
  - DLRM recommendation system









Ke et al. "Near-Memory Processing in Action: Accelerating Personalized Recommendation with AxDIMM", IEEE Micro (2021)

SK Hynix Accelerator-in-Memory (2022)

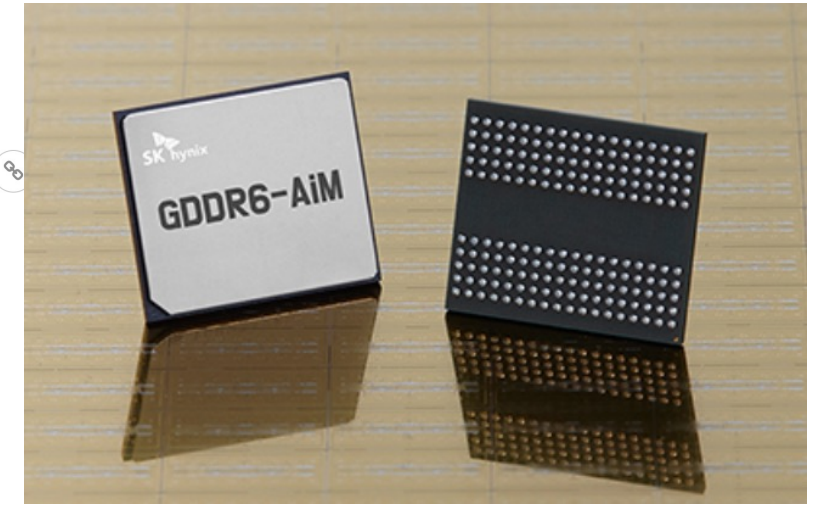
SKhynix NEWSROOM

ENG

INSIGHT | SK hynix STORY | PRESS CENTER | MULTIMEDIA | Search

SK hynix Develops PIM, Next-Generation AI Accelerator

February 16, 2022



Seoul, February 16, 2022

SK hynix (or "the Company", www.skhynix.com) announced on February 16 that it has developed PIM*, a next-generation memory chip with computing capabilities.

*PIM(Processing In Memory): A next-generation technology that provides a solution for data congestion issues for AI and big data by adding computational functions to semiconductor memory

It has been generally accepted that memory chips store data and CPU or GPU, like human brain, process data. SK hynix, following its challenge to such notion and efforts to pursue innovation in the next-generation smart memory, has found a breakthrough solution with the development of the latest technology.

SK hynix plans to showcase its PIM development at the world's most prestigious semiconductor conference, 2022 ISSCC*, in San Francisco at the end of this month. The company expects continued efforts for innovation of this technology to bring the memory-centric computing, in which semiconductor memory plays a central role, a step closer to the reality in devices such as smartphones.

*ISSCC: The International Solid-State Circuits Conference will be held virtually from Feb. 20 to Feb. 24 this year with a theme of "Intelligent Silicon for a Sustainable World"

For the first product that adopts the PIM technology, SK hynix has developed a sample of GDDR6-AiM (Accelerator* in memory). The GDDR6-AiM adds computational functions to GDDR6* memory chips, which process data at 16Gbps. A combination of GDDR6-AiM with CPU or GPU instead of a typical DRAM makes certain computation speed 16 times faster. GDDR6-AiM is widely expected to be adopted for machine learning, high-performance computing, and big data computation and storage.

11.1 A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications

Seongju Lee, SK hynix, Icheon, Korea

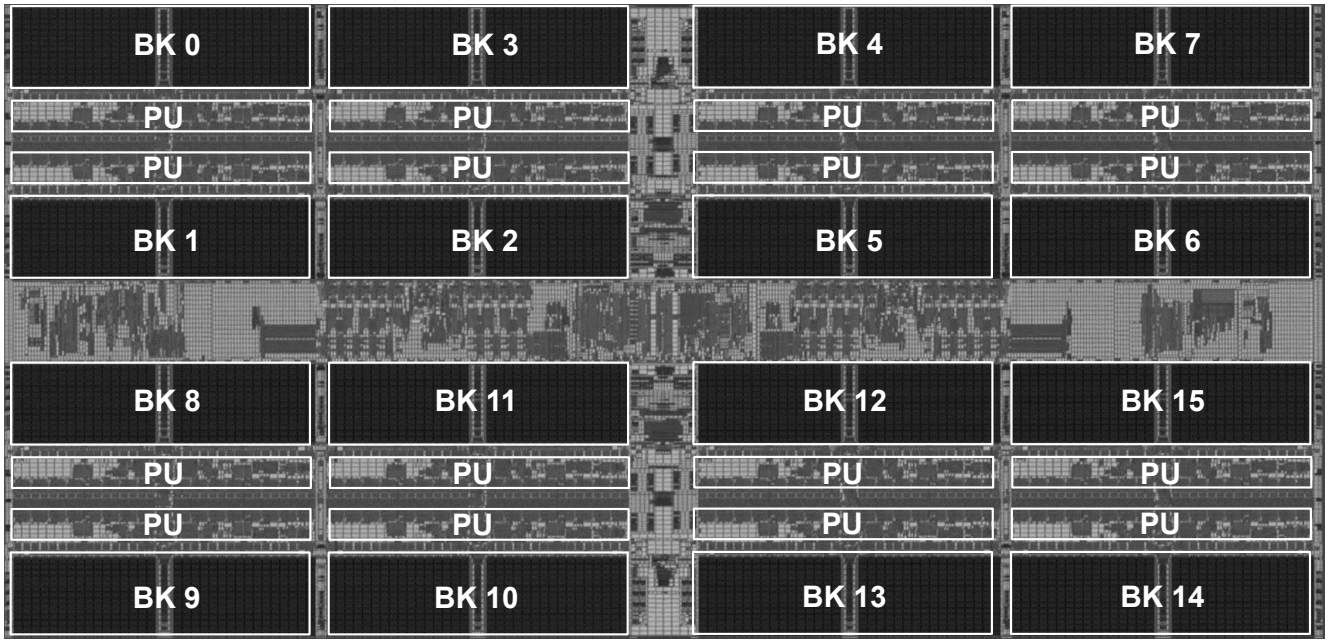
In Paper 11.1, SK Hynix describes an 1ynm, GDDR6-based accelerator-in-memory with a command set for deep-learning operation. The 8Gb design achieves a peak throughput of 1TFLOPS with 1GHz MAC operations and supports major activation functions to improve accuracy.

https://news.skhynix.com/sk-hynix-develops-pim-next-generation-ai-accelerator/

SK Hynix AiM: Chip Implementation (2022)

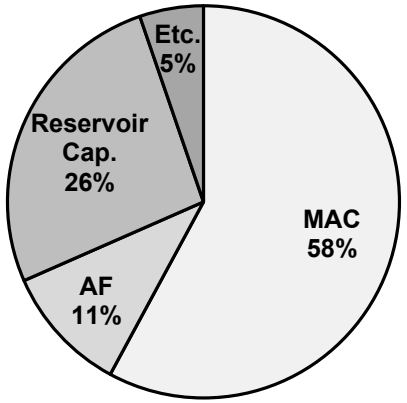
- 4 Gb AiM die with 16 processing units (PUs)

AiM Die Photograph



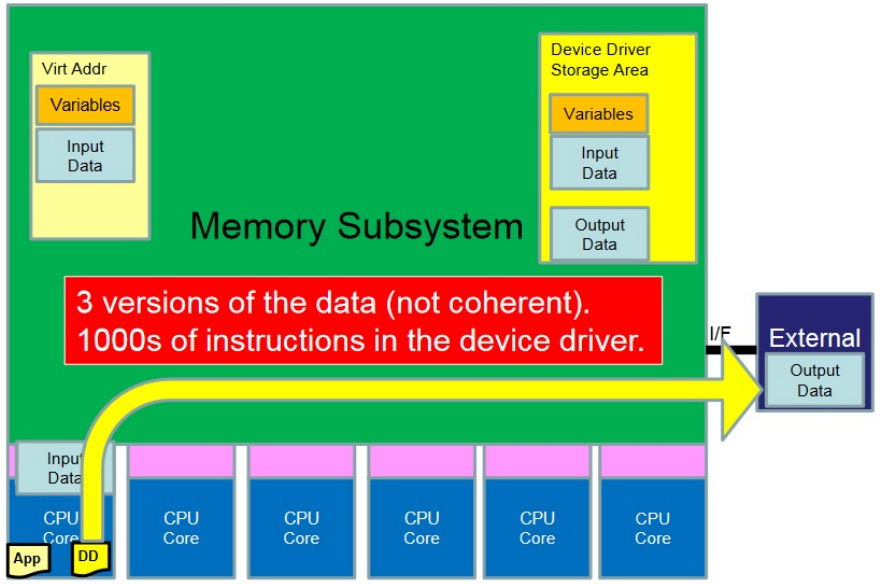
1 Process Unit (PU) Area

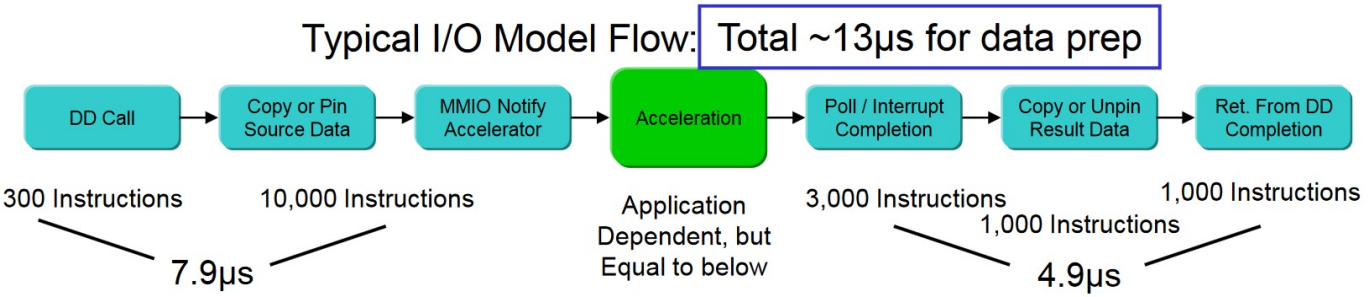
Total	0.19mm^2
MAC	0.11mm^2
Activation Function (AF)	0.02mm^2
Reservoir Cap.	0.05mm^2
Etc.	0.01mm^2



Lee et al., A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications, ISSCC 2022

Background: Traditional I/O Technology

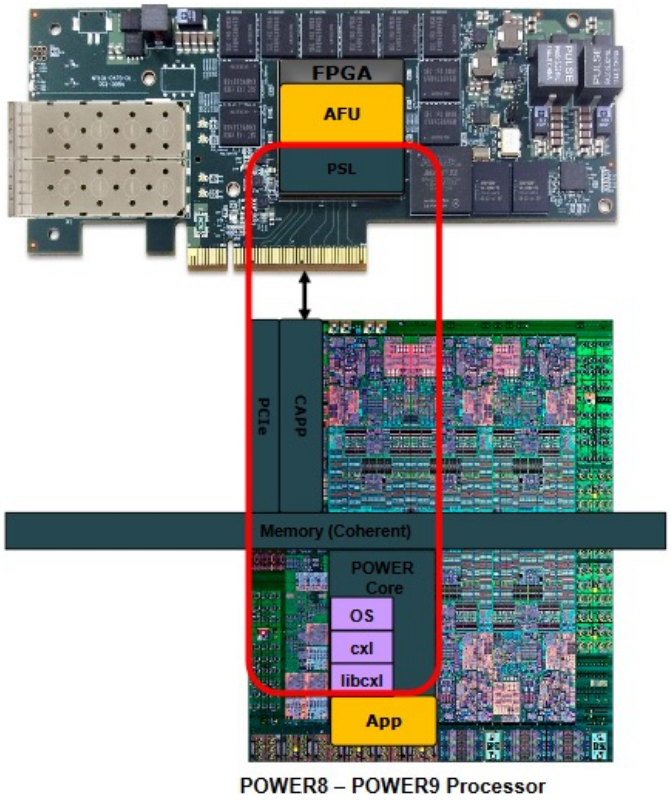




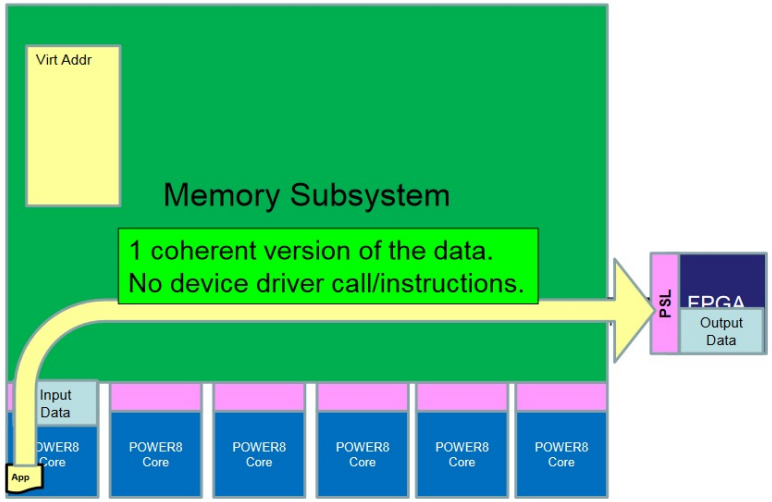
Dionysios Diamantopoulos, IBM Research – Zurich, COOL Chips 2018

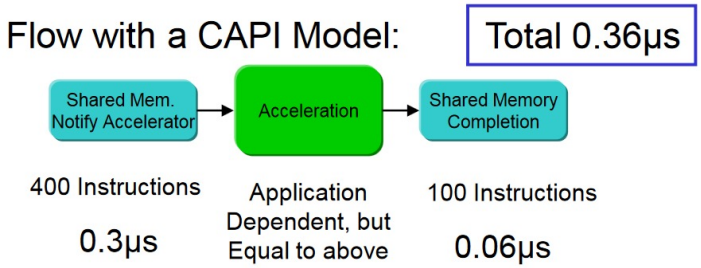
CAPI/OpenCAPI Overview

- CAPI/CAPI2 (Coherent Accelerator Processor Interface)
- OpenCAPI



POWER8 – POWER9 Processor



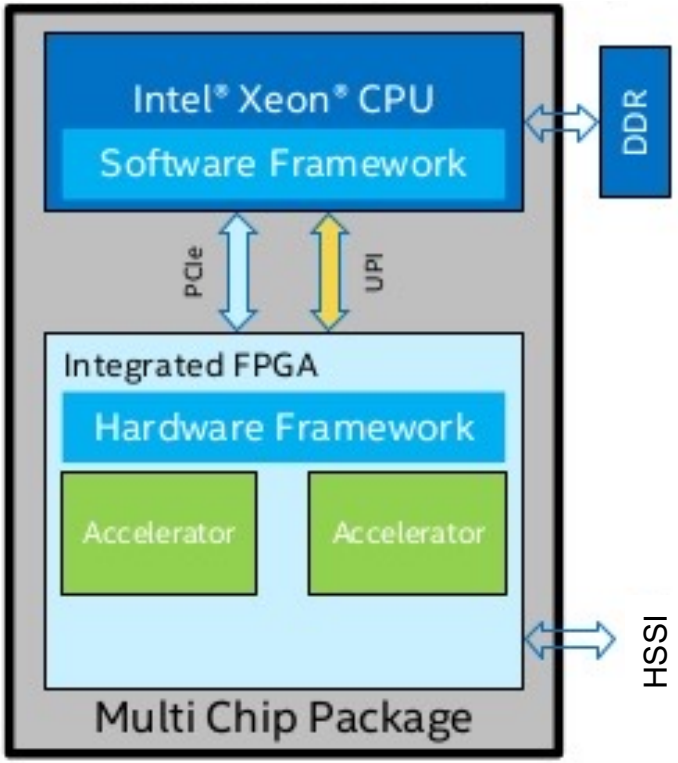


Dionysios Diamantopoulos, IBM Research – Zurich, COOL Chips 2018

Collaborative Computing on CPU+FPGA

- Traditionally, accelerators (GPUs, FPGAs, etc.) have been used as *offload* engines
- Heterogeneous architectures moving towards tighter integration
  - Unified memory
  - System-wide atomics
- Tighter integration allows fine-grained collaboration

Key challenge: identify the best CPU-FPGA collaboration strategy

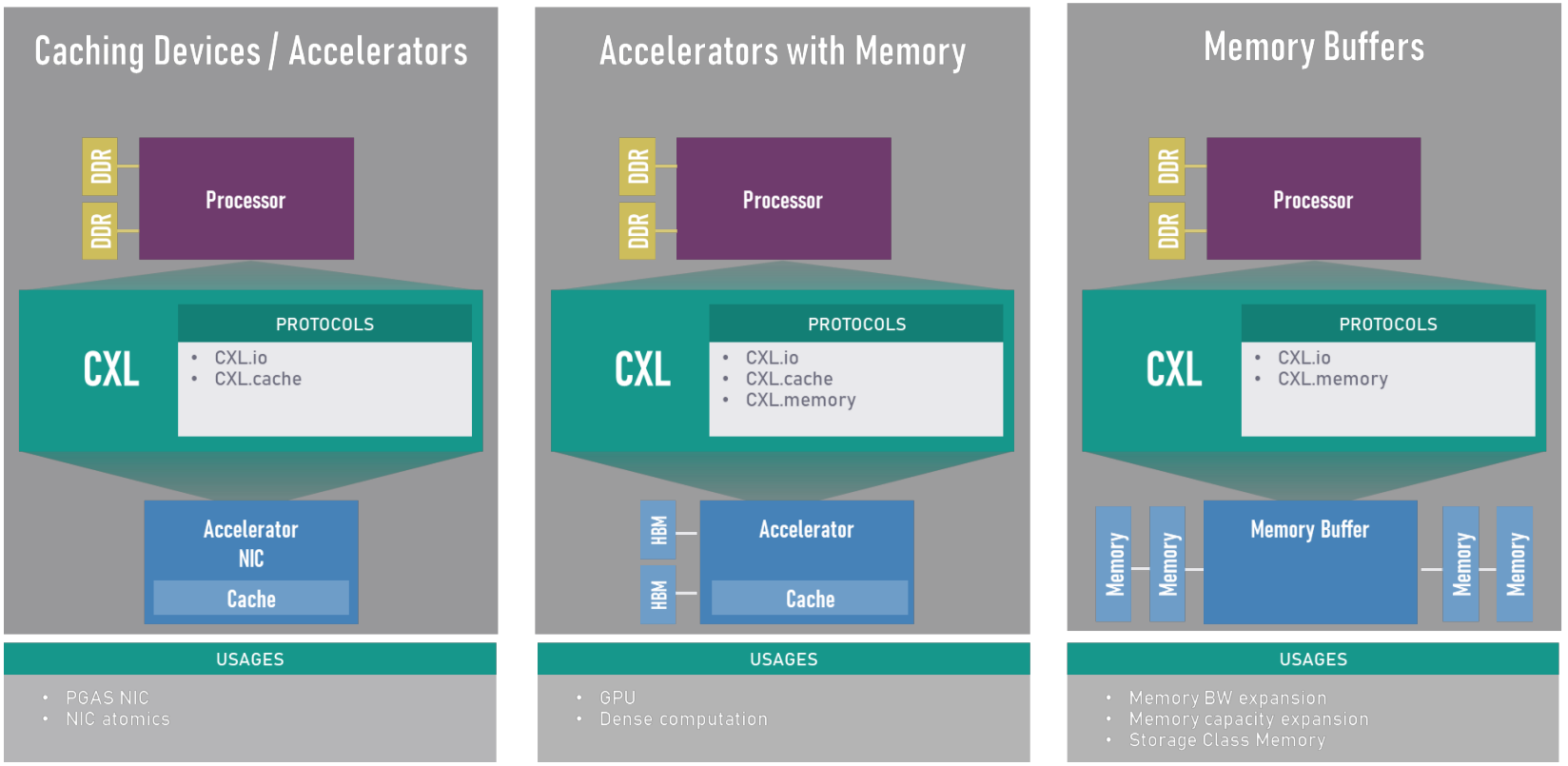


Intel Xeon + FPGA Integrated Platform (MCP)

Huang+, "Analysis and Modeling of Collaborative Execution Strategies for Heterogeneous CPU-FPGA Architectures," ICPE 2019

Compute Express Link (CXL)

- Compute Express Link (CXL) is an open industry standard interconnect offering high-bandwidth, low-latency connectivity between host processor and devices such as accelerators, memory buffers, and smart I/O devices



https://www.computeexpresslink.org/_files/ugd/0c1418_14c5283e7f3e40f9b2955c7d0f60bebe.pdf

Key Takeaways

- This P&S is aimed at improving your
  - Knowledge in Computer Architecture and Heterogeneous Systems
  - Technical skills in programming heterogeneous architectures
  - Critical thinking and analysis
  - Interaction with a nice group of researchers
  - Familiarity with key research directions
  - Technical presentation of your project

Key Goal

(Learn how to) take advantage of existing heterogeneous devices by programming them, analyzing workloads, proposing offloading/scheduling techniques…

Prerequisites of the Course

- Digital Design and Computer Architecture (or equivalent course)
  - https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=schedule
  - https://safari.ethz.ch/digitaltechnik/spring2022/doku.php?id=schedule
- Familiarity with C/C++ programming
  - FPGA implementation or GPU programming (desirable)
- Interest in
  - computer architectures and computing paradigms
  - discovering why things do or do not work and solving problems
  - making systems efficient and usable

Course Info: Who Are We? (I)

- Onur Mutlu
  - Full Professor @ ETH Zurich ITET (INFK), since September 2015
  - Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-…
  - PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
  - https://people.inf.ethz.ch/omutlu/
  - omutlu@gmail.com (Best way to reach me)
  - https://people.inf.ethz.ch/omutlu/projects.htm
- Research and Teaching in:
  - Computer architecture, computer systems, hardware security, bioinformatics
  - Memory and storage systems
  - Hardware security, safety, predictability
  - Fault tolerance
  - Hardware/software cooperation
  - Architectures for bioinformatics, health, medicine
  - …

Course Info: Who Are We? (II)

- Lead Supervisor:
  - Dr. Juan Gómez Luna

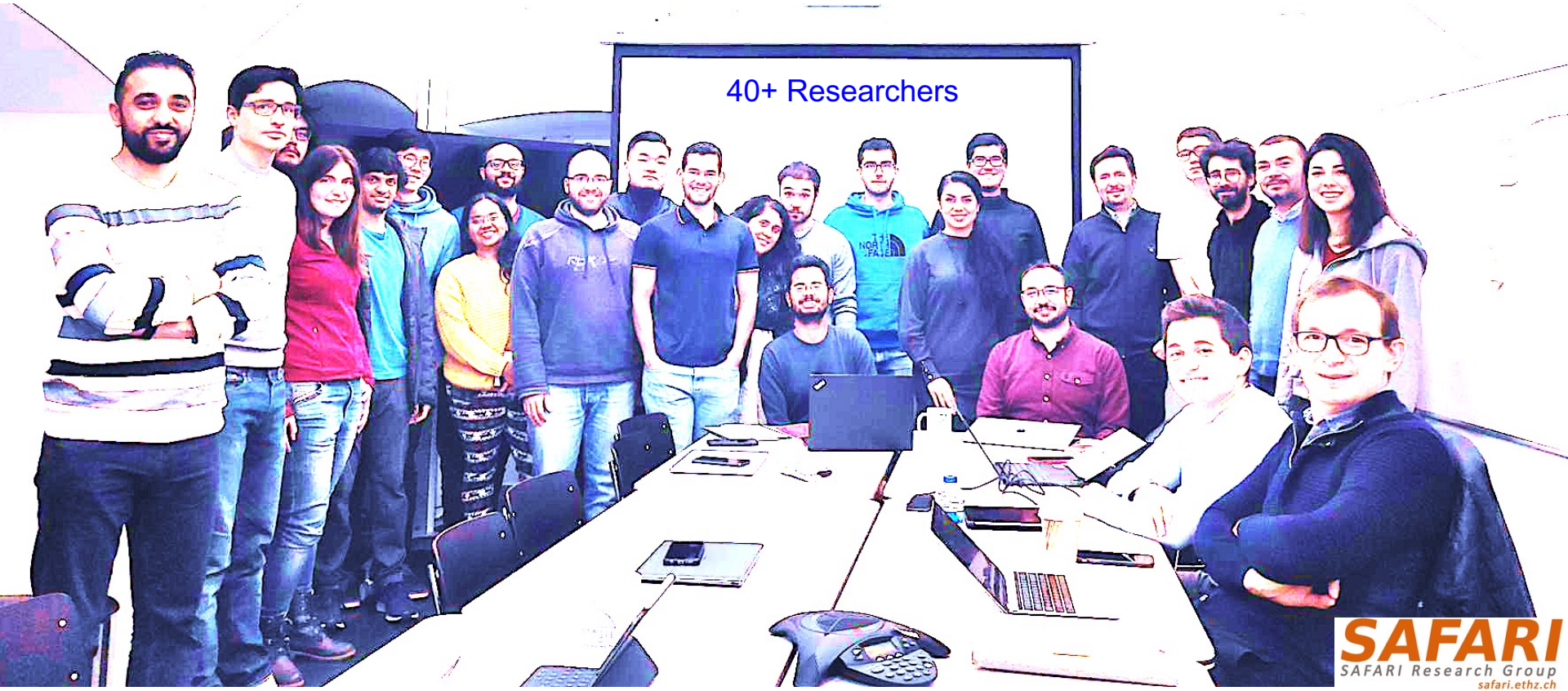
- Supervisors:
  - Dr. Mohammed Alser
  - Dr. Behzad Salami
  - Dr. Mohammad Sadr
  - Joel Lindegger

- Get to know us and our research
  - https://safari.ethz.ch/safari-group/

Onur Mutlu's SAFARI Research Group

Computer architecture, HW/SW, systems, bioinformatics, security, memory

https://safari.ethz.ch/safari-newsletter-january-2021/



Think BIG, Aim HIGH!

SAFARI Newsletter December 2021 Edition

- https://safari.ethz.ch/safari-newsletter-december-2021/

SAFARI
SAFARI Research Group

Think Big, Aim High

ETH zürich

View in your browser
December 2021

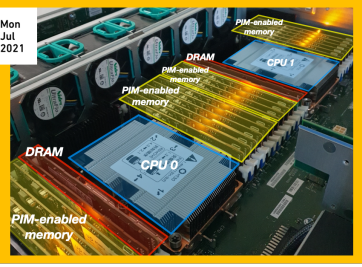
SAFARI Live Seminars (I)

SAFARI Live Seminars in Computer Architecture

Dr. Juan Gómez Luna, ETH Zurich

Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

12 Mon Jul 2021

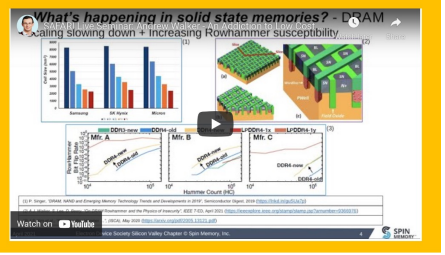


SAFARI Live Seminars in Computer Architecture

Dr. Andrew Walker, Schiltron Corporation & Nexgen Power Systems

An Addiction to Low Cost Per Memory Bit – How to Recognize it and What to Do About it

19 Mo Jul 2021

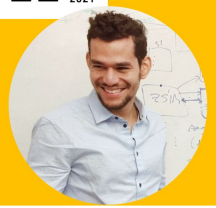


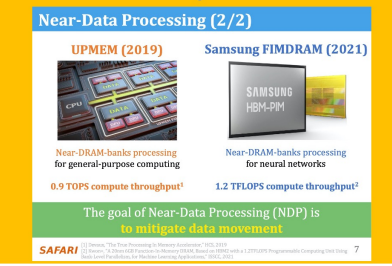
SAFARI Live Seminars in Computer Architecture

Geraldo F. Oliveira, ETH Zurich

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

22 Do Jul 2021



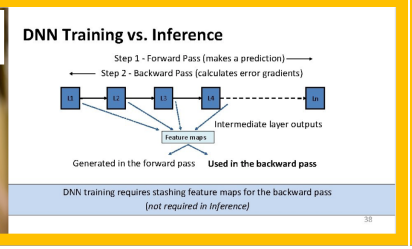


SAFARI Live Seminars in Computer Architecture

Gennady Pekhimenko, University of Toronto

Efficient DNN Training at Scale: from Algorithms to Hardware

5 Do Aug 2021

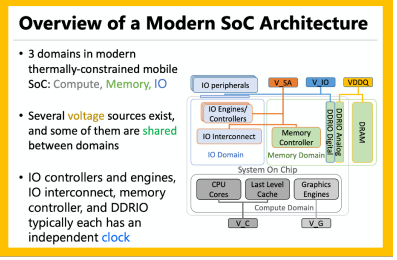


SAFARI Live Seminars in Computer Architecture

Jawad Haj-Yahya, Huawei Research Center Zurich

Power Management Mechanisms in Modern Microprocessors and Their Security Implications

16 Mo Aug 2021

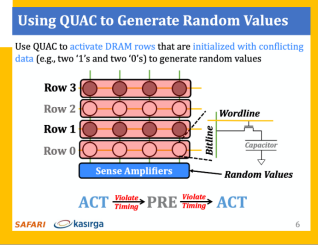


SAFARI Live Seminars in Computer Architecture

Ataberk Olgun, TOBB & ETH Zurich

QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips

15 Mi Sep 2021

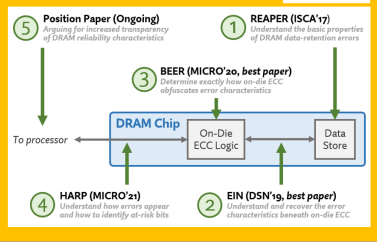


SAFARI Live Seminars in Computer Architecture

Minesh Patel, ETH Zurich

Enabling Effective Error Mitigation in Memory Chips That Use On-Die ECCs

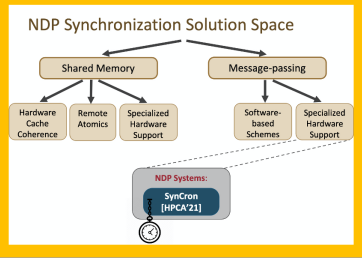
21 Tues Sep 2021



SAFARI Live Seminars in Computer Architecture

Christina Giannoula, National Technical University of Athens

Efficient Synchronization Support for Near-Data-Processing Architectures



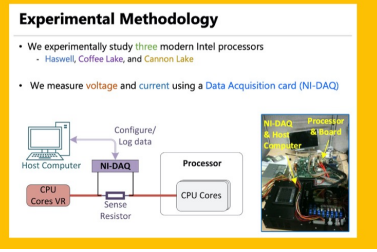
27 Mo Sep 2021

SAFARI Live Seminars in Computer Architecture

Jawad Haj-Yahya, Huawei Research Center Zurich

Security Implications of Power Management Mechanisms In Modern Processors, Current Studies and Future Trends

4 Mo Okt 2021

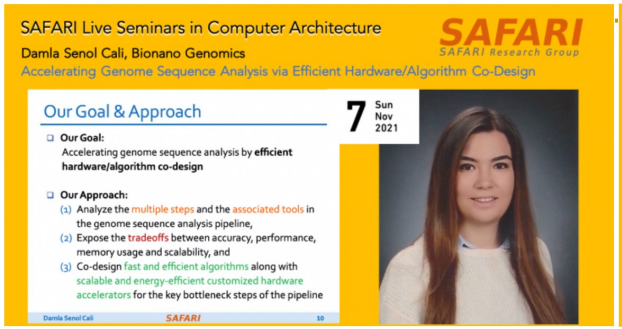


https://safari.ethz.ch/safari-seminar-series/

SAFARI Live Seminars (II)

SAFARI Live Seminar: Nastaran Hajinazar 27 Oct 2021

Posted on October 1, 2021 by ewent



SAFARI Live Seminar: Damla Senol Cali 07 Nov 2021

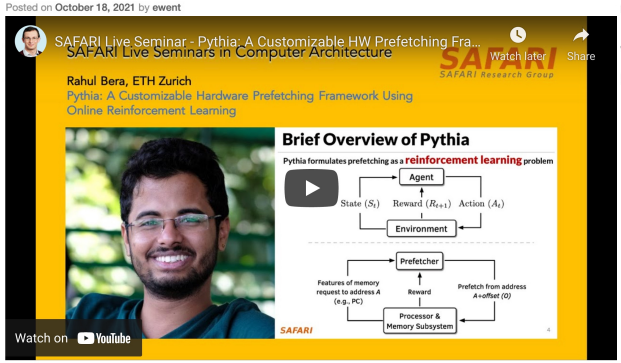
Posted on October 18, 2021 by ewent

SAFARI Live Seminar: Gennady Pekhimenko 08 Nov 2021

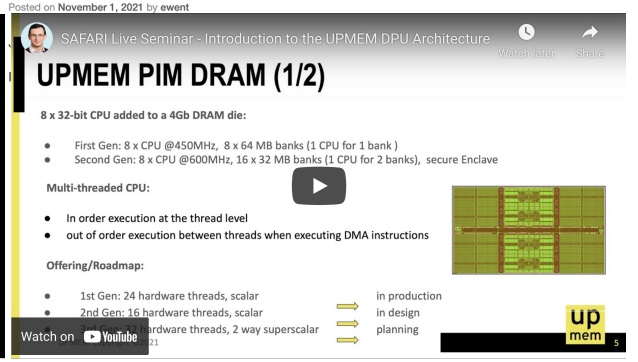
Posted on November 1, 2021 by ewent

SAFARI Live Seminar: Serghei Mangul 11 Nov 2021

Posted on November 5, 2021 by ewent



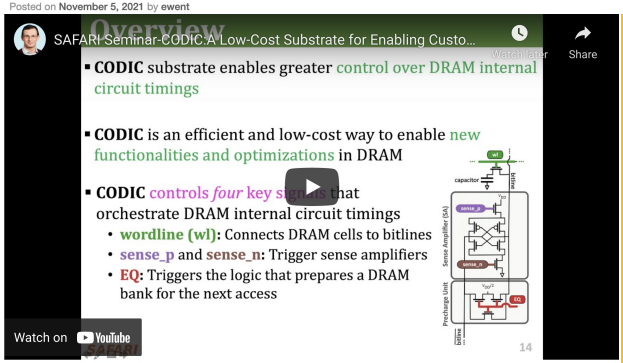
SAFARI Live Seminar: Rahul Bera 20 Dec 2021



SAFARI Live Seminar: Fabrice Devaux, 2 Feb 2022

Posted on January 15, 2022 by ewent

Join us for our joint SAFARI Live Seminar & EFCL Seminar with Fabrice Devaux, UPMEM
Wednesday, February 2 2022 at 11:00 am Zurich time (CET)



SAFARI Live Seminar: Lois Orosa, 10 Feb 2022

Posted on January 16, 2022 by ewent

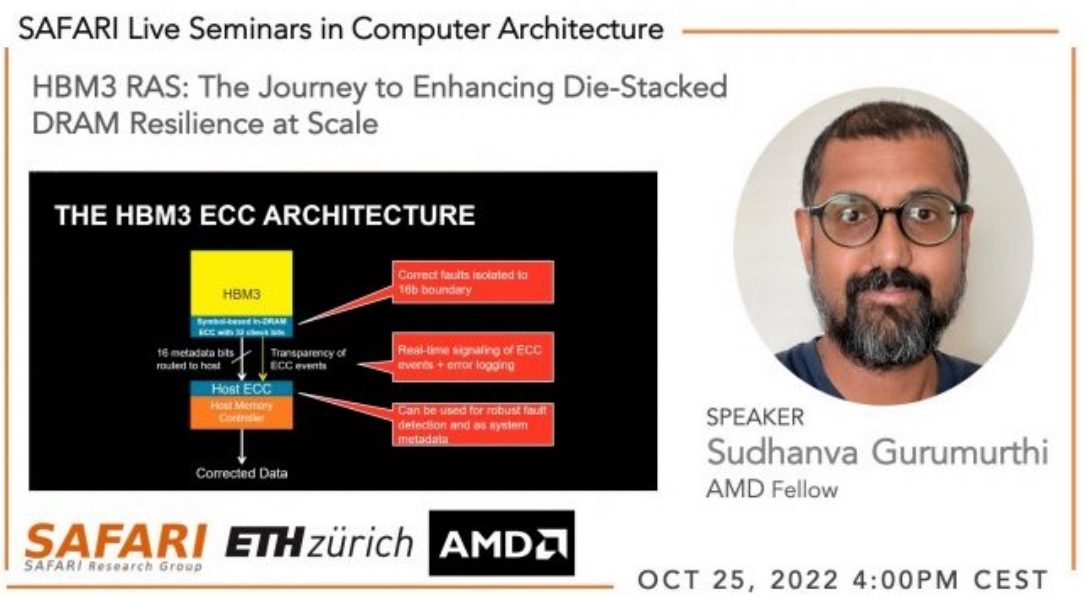
Join us for our next SAFARI Live Seminar with Lois Orosa.
Thursday, February 10 at 5:00 pm Zurich time (CET)

Posted on January 19, 2022 by ewent

Join us for our SAFARI Live Seminar with Sean Lie, Cerebras Systems
Monday, February 28 2022 at 6:00 pm Zurich time (CET)

https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9&index=1

SAFARI Live Seminars (Upcoming Talks)



SAFARI Live Seminar: Sudhanva Gurumurthi, Oct 25 2022

Posted on **September 6, 2022** by **ewent**

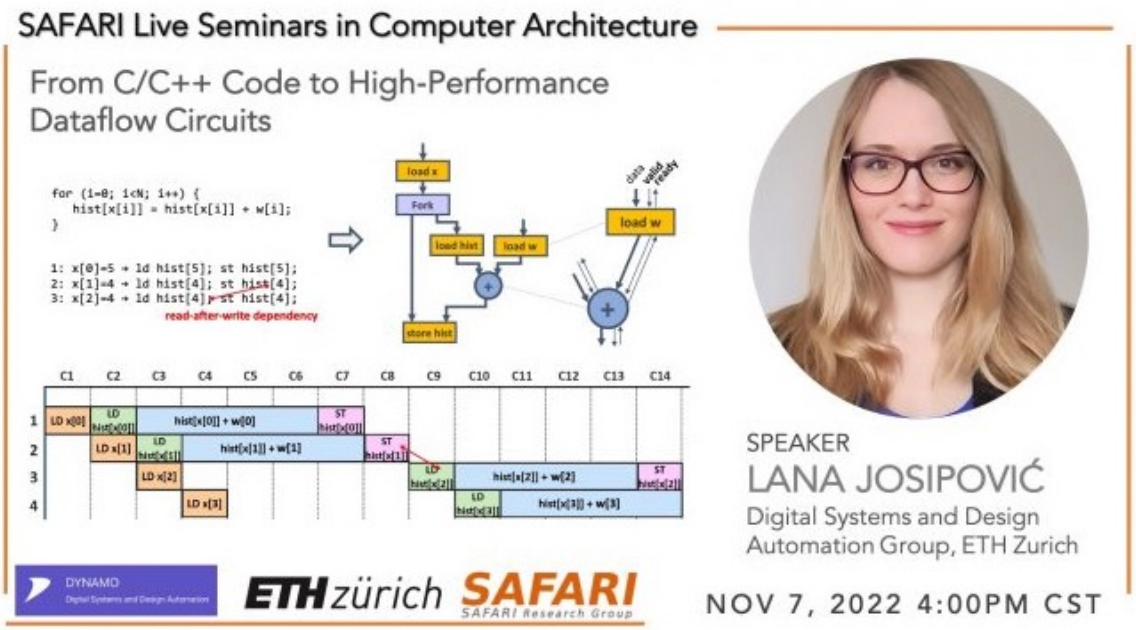
We're excited to have Sudhanva Gurumurthi with us for our upcoming SAFARI Live Seminar!

Date: Tuesday, October 25 at 4:00 pm Zurich time (CEST)
Speaker: Sudhanva Gurumurthi, AMD Fellow
Link: Livestream on YouTube Link

Title: HBM3 RAS: The Journey to Enhancing Die-Stacked DRAM Resilience at Scale

https://safari.ethz.ch/safari-seminar-series/

SAFARI Live Seminars (Upcoming Talks)



SAFARI Live Seminar: Lana Josipović, Nov 7 2022

Posted on **September 9, 2022** by **ewent**

Join us for our upcoming SAFARI Live Seminar

Date: Monday, November 7 at 4:00 pm Zurich time (CET)

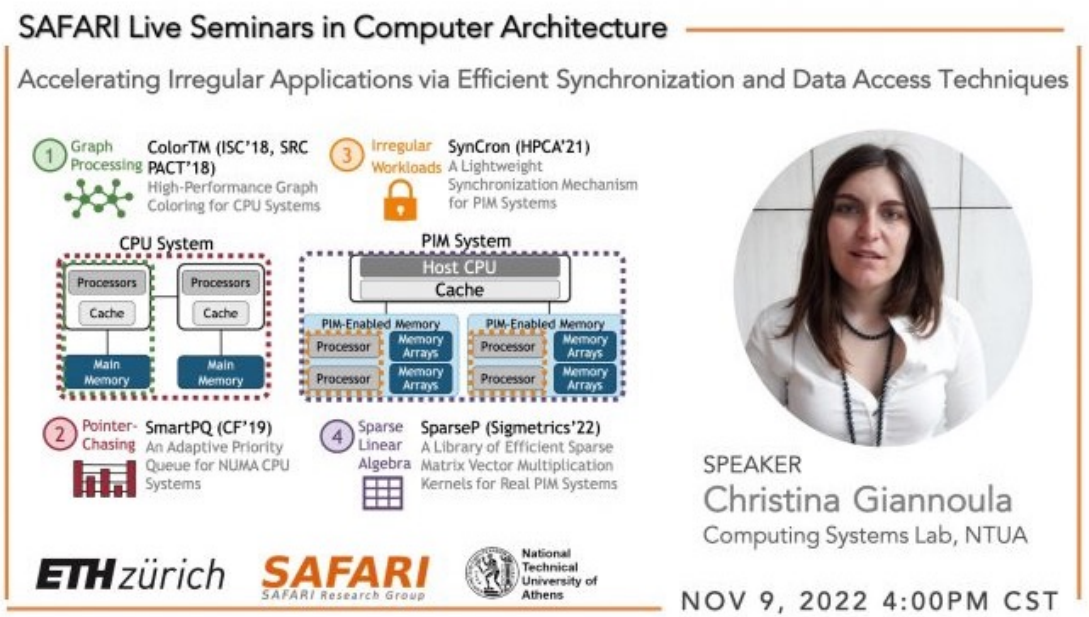
Speaker: Lana Josipovic, DYNAMO Research Group, ETH Zurich

Link: Livestream on YouTube Link

Title: From C/C++ Code to High-Performance Dataflow Circuits

https://safari.ethz.ch/safari-seminar-series/

SAFARI Live Seminars (Upcoming Talks)



SAFARI Live Seminar, Christina Giannoula, Nov 9 2022

Posted on **September 15, 2022** by **ewent**

Join us for our upcoming SAFARI Live Seminar

Date: Wednesday, November 9 at 4:00 pm Zurich time (CET)

Speaker: Christina Giannoula, School of Electrical and Computer Engineering, NTUA

Link: Livestream on YouTube Link

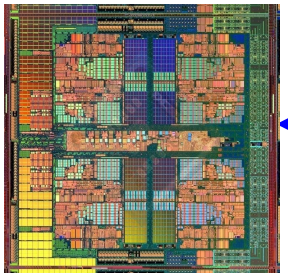
Title: Accelerating Irregular Applications via Efficient Synchronization and Data Access Techniques

https://safari.ethz.ch/safari-seminar-series/

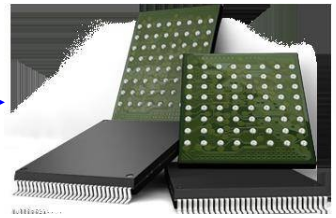
Current Research Focus Areas

***Research Focus:* Computer architecture, HW/SW, bioinformatics**

- *Memory and storage (DRAM, flash, emerging), interconnects*
- *Heterogeneous & parallel systems, GPUs, systems for data analytics*
- *System/architecture interaction, new execution models, new interfaces*
- *Energy efficiency, fault tolerance, hardware security, performance*
- *Genome sequence analysis & assembly algorithms and architectures*
- *Biologically inspired systems & system design for bio/medicine*

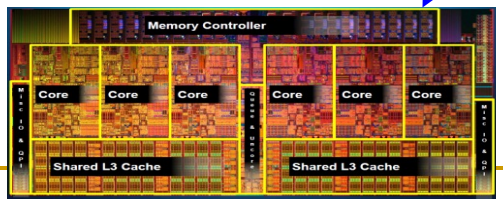


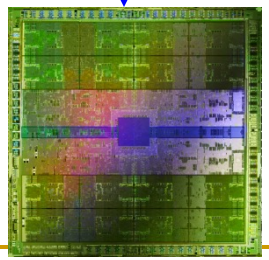
Hybrid Main Memory



Persistent Memory/Storage

Heterogeneous Processors and Accelerators





Graphics and Vision Processing

Broad research spanning apps, systems, logic with architecture at the center

Course Info: How About You?

- Let us know your background, interests
- Why did you join this P&S?

Course Requirements and Expectations

- Attendance required for all meetings
- Study the learning materials
- Each student will carry out a hands-on project
  - Build, implement, code, and design with close engagement from the supervisors
- Participation
  - Ask questions, contribute thoughts/ideas
  - Read relevant papers

We will help in all projects!

If your work is really good, you may get it published!

Course Website

- https://safari.ethz.ch/projects_and_seminars/doku.php?id=heterogeneous_systems
- Useful information about the course
- Check your email frequently for announcements
- We also have Moodle for Q&A

Meeting 1

- Recommended materials:

1. An introduction to SIMD processors and GPUs (Dr. Juan Gomez Luna, lecture).
(PDF) (PPT) Video

2. An introduction to GPUs and heterogeneous programming (Dr. Juan Gomez Luna, lecture).
(PDF) (PPT) Video

- Other recommended materials:

3. Juan Gomez-Luna, Izzat El Hajj, Li-Wen Chang, Victor Garcia-Flores, Simon Garcia de Gonzalo, Thomas B. Jablin, Antonio J. Peña and Wen-mei Hwu,
"Chai: Collaborative Heterogeneous Applications for Integrated-architectures"
Proceedings of the 2017 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Santa Rosa, California, April 2017.
https://chai-benchmarks.github.io https://github.com/chai-benchmarks/chai

4. Gagandeep Singh, Dionysios Diamantopoulos, Christoph Hagleitner, Juan Gómez-Luna, Sander Stuijk, Onur Mutlu, and Henk Corporaal,
"NERO: A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling"
Proceedings of the 30th International Conference on Field-Programmable Logic and Applications (FPL), Gothenburg, Sweden, September 2020.
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (23 minutes)]

5. Mohammed Alser, Taha Shahroodi, Juan Gomez Luna, Can Alkan, and Onur Mutlu,
"SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs"
Bioinformatics, 26 December 2020.
[Source Code] [Online link at Bioinformatics Journal]

6. Real Processing-in-DRAM with UPMEM (Dr. Juan Gomez Luna, SAFARI Live Seminar, July 2021).
"Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture"
Preprint in arXiv, 9 May 2021.
[PrIM Benchmarks Source Code]
[Slides (pptx) (pdf)]
[SAFARI Live Seminar Slides (pptx) (pdf)]
[SAFARI Live Seminar Video (2 hrs 57 mins)]

Meeting 2

- We will announce the projects and will give you some description about them
- We will give you a chance to select a project
- Then, we will have 1-1 meetings to match your interests, skills, and background with a suitable project
- It is important that you study the learning materials before our next meeting!

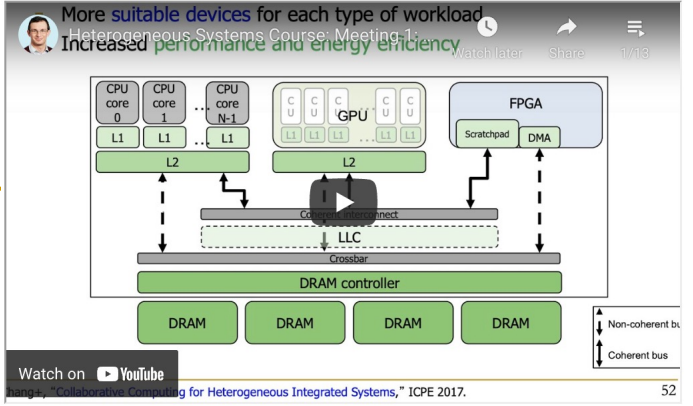
Next Meetings

- Individual meetings with your mentor/s
- Tutorials and short talks
  - GPU programming
  - Recent research works
- Presentation of your work

Hetero. Systems (Spring 2022)

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=heterogeneous_systems
- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=oFO5fTrgFIY&list=PL5Q2soXY2Zi9XrgXR38IM_FTjmY6h7Gzm
- Project course
  - Taken by Bachelor's/Master's students
  - GPU and Parallelism lectures
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures



Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	15.03 Tue.	Premiere	M1: P&S Course Presentation (PDF) (PPT)	Required Materials Recommended Materials	HW 0 Out
W2	22.03 Tue.	Premiere	M2: SIMD Processing and GPUs (PDF) (PPT)		
W3	29.03 Tue.	Premiere	M3: GPU Software Hierarchy (PDF) (PPT)		
W4	05.04 Tue.	Premiere	M4: GPU Memory Hierarchy (PDF) (PPT)		
W5	12.04 Tue.	Premiere	M5: GPU Performance Considerations (PDF) (PPT)		
W6	19.04 Tue.	Premiere	M6: Parallel Patterns: Reduction (PDF) (PPT)		
W7	26.04 Tue.	Premiere	M7: Parallel Patterns: Histogram (PDF) (PPT)		
W8	03.05 Tue.	Premiere	M8: Parallel Patterns: Convolution (PDF) (PPT)		
W9	10.05 Tue.	Premiere	M9: Parallel Patterns: Prefix Sum (Scan) (PDF) (PPT)		
W10	17.05 Tue.	Premiere	M10: Parallel Patterns: Sparse Matrices (PDF) (PPT)		
W11	24.05 Tue.	Premiere	M11: Parallel Patterns: Graph Search (PDF) (PPT)		
W12	01.06 Wed.	Premiere	M12: Parallel Patterns: Merge Sort (PDF) (PPT)		
W13	07.06 Tue.	Premiere	M13: Dynamic Parallelism (PDF) (PPT)		
W14	15.06 Wed.	Premiere	M14: Collaborative Computing (PDF) (PPT)		
W15	24.06 Fri.	Premiere	M15: GPU Acceleration of Genome Sequence Alignment (PDF) (PPT)		
W16	14.07 Thu.	Premiere	M16: Accelerating Agent-based Simulations (PDF) (ODP)		

Exploiting Data Parallelism: SIMD Processors and GPUs

Recall: Flynn's Taxonomy of Computers

- Mike Flynn, "Very High-Speed Computing Systems," Proc. of IEEE, 1966

- SISD: Single instruction operates on single data element
- SIMD: Single instruction operates on multiple data elements
  - Array processor
  - Vector processor
- MISD: Multiple instructions operate on single data element
  - Closest form: systolic array processor, streaming processor
- MIMD: Multiple instructions operate on multiple data elements (multiple instruction streams)
  - Multiprocessor
  - Multithreaded processor

Recall: MMX Example: Image Overlaying (I)

- Goal: Overlay the human in image x on top of the background in image y

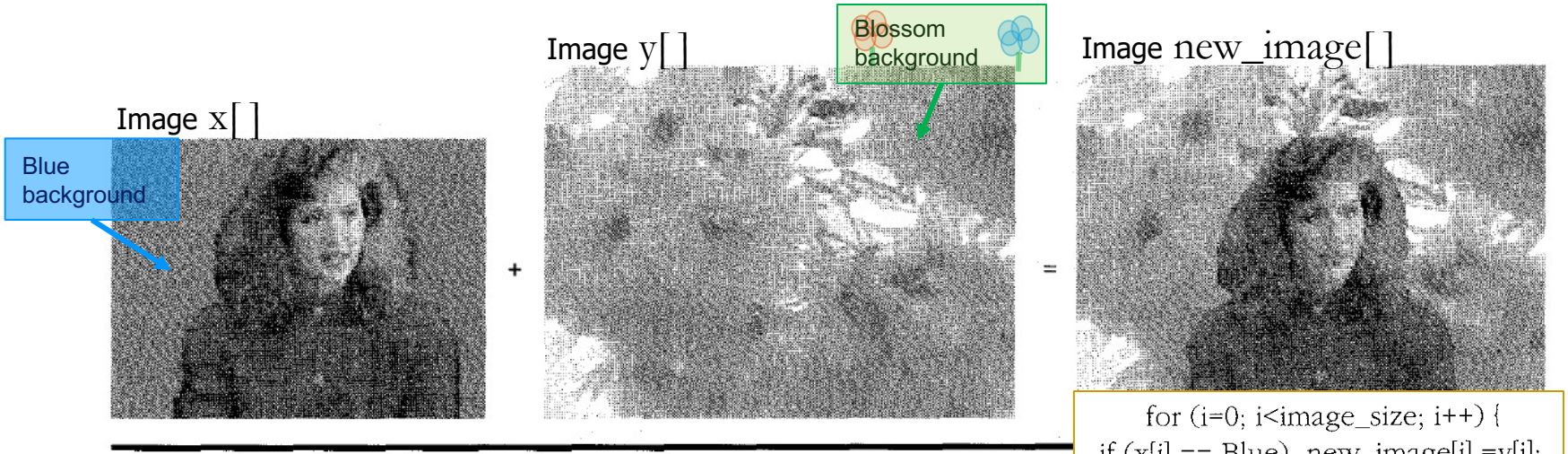


Figure 8. Chroma keying: image overlay using a background color.

```
for (i=0; i<image_size; i++) {
if (x[i] == Blue)  new_image[i] =y[i];
        else new_image[i] = x[i];
}
```

PCMPEQB MM1, MM3

MM1	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue
Image x[ ] MM3	X7!=blue	X6!=blue	X5=blue	X4=blue	X3!=blue	X2!=blue	X1=blue	X0=blue
Bit mask MM1	0x0000	0x0000	0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF

Bitmask

Figure 9. Generating the selection bit mask.

SIMD Processing

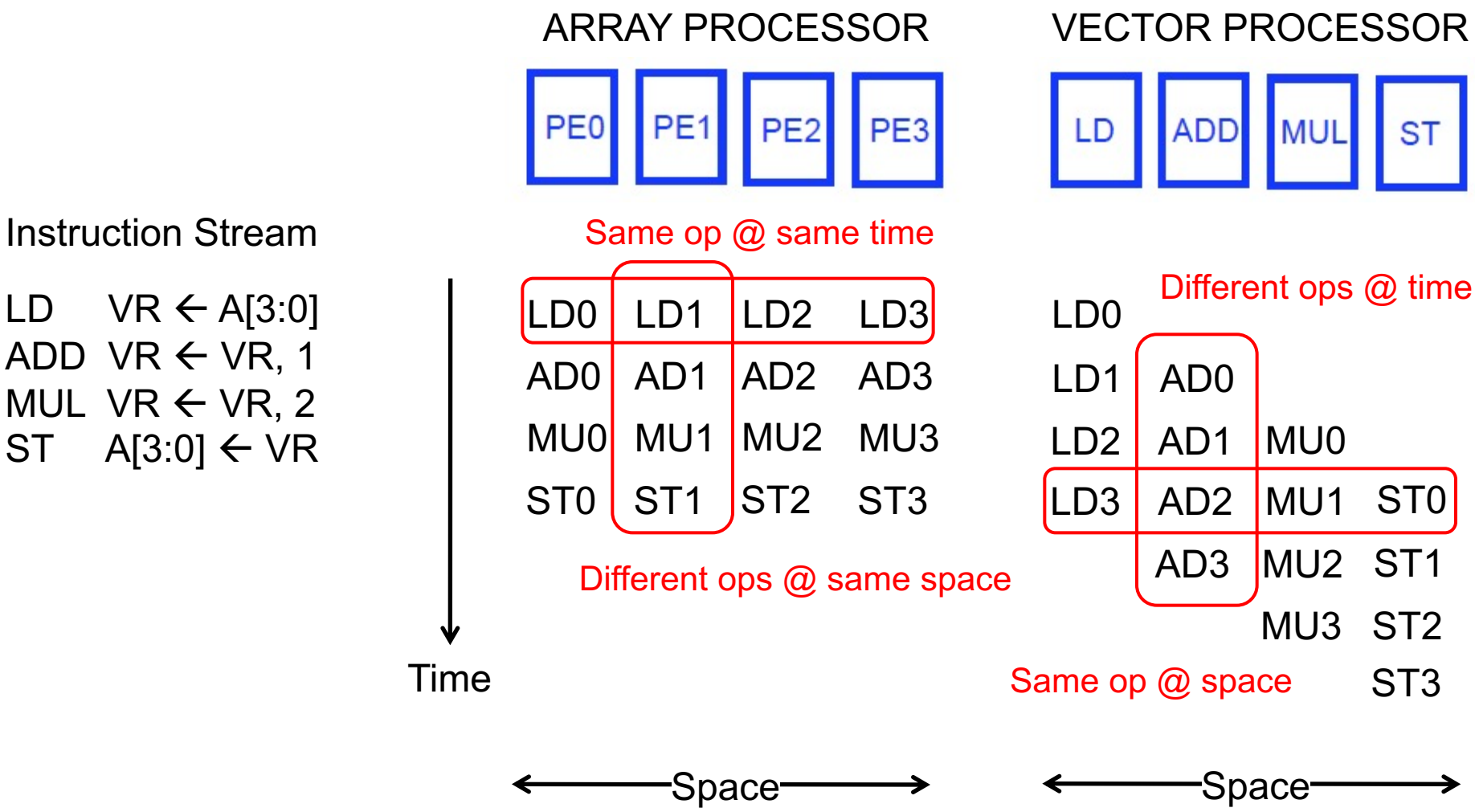
- Single instruction operates on multiple data elements
  - In time or in space
- Multiple processing elements (PEs), i.e., execution units

- Time-space duality

  - Array processor: Instruction operates on multiple data elements at the same time using different spaces (PEs)

  - Vector processor: Instruction operates on multiple data elements in consecutive time steps using the same space (PE)

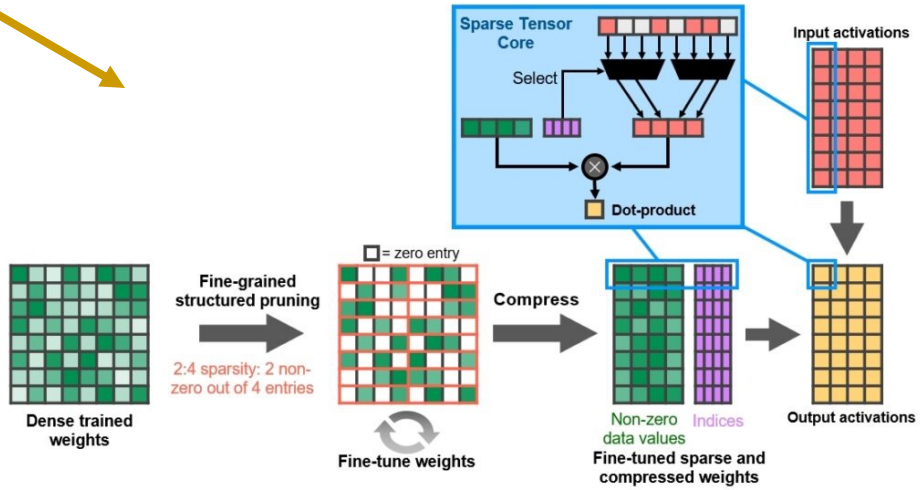
Array vs. Vector Processors



NVIDIA A100 Core

GPU compute throughput:

- 19.5 TFLOPS Single Precision
- 9.7 TFLOPS Double Precision
- 312 TFLOPS for Deep Learning (Tensor cores)



https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

Vector Processor Disadvantages

-- Works (only) if parallelism is regular (data/SIMD parallelism)

++ Vector operations

-- Very inefficient if parallelism is irregular

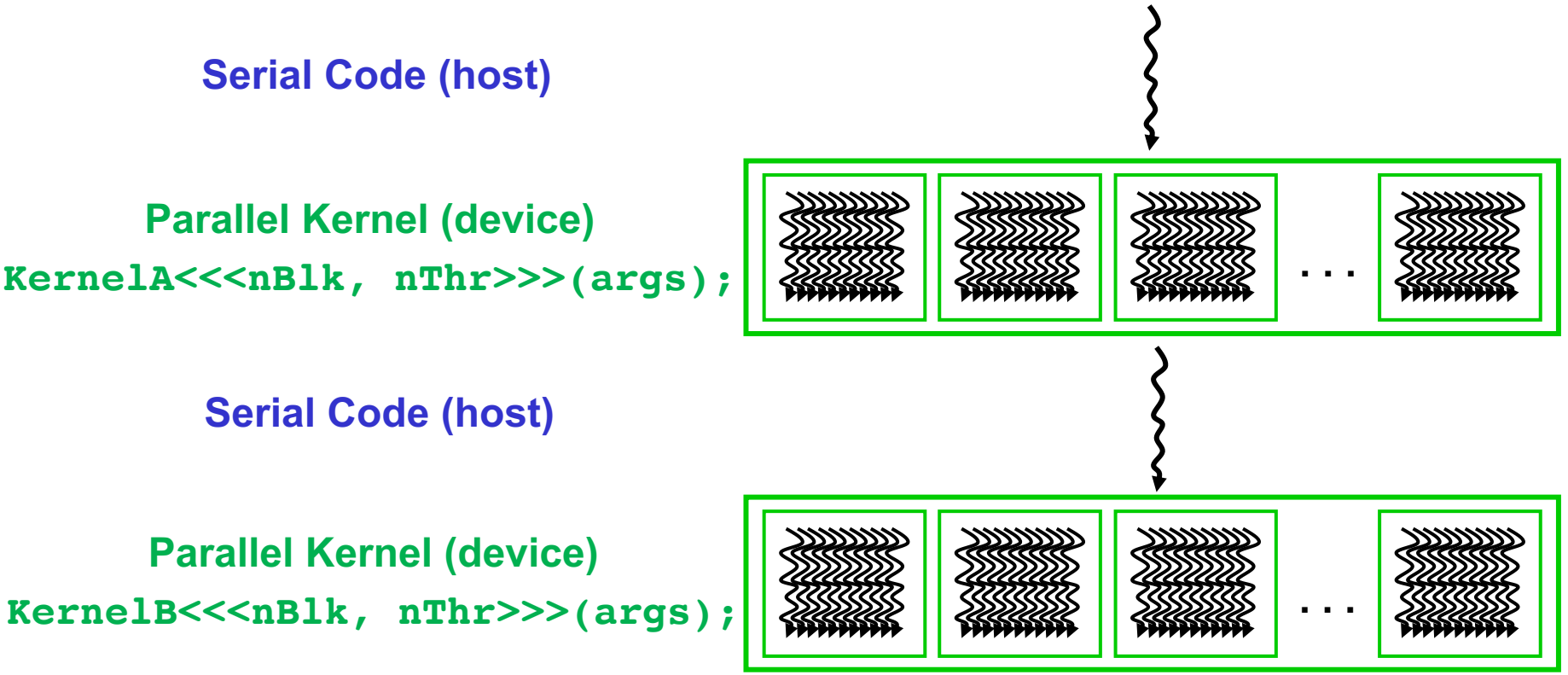
-- How about searching for a key in a linked list?

To program a vector machine, the compiler or hand coder must make the data structures in the code fit nearly exactly the regular structure built into the hardware. That's hard to do in first place, and just as hard to change. One tweak, and the low-level code has to be rewritten by a very smart and dedicated programmer who knows the hardware and often the subtleties of the application area. Often the rewriting is

Fisher, "Very Long Instruction Word architectures and the ELI-512," ISCA 1983.

Warps *not* Exposed to GPU Programmers

- CPU threads and GPU kernels
  - Sequential or modestly parallel sections on CPU
  - Massively parallel sections on GPU: Blocks of threads



Slide credit: Hwu & Kirk

Sample GPU SIMT Code (Simplified)

CPU code

```
for (ii = 0; ii < 100000; ++ii) {
C[ii] = A[ii] + B[ii];
}
```

CUDA code

```
// there are 100000 threads
__global__ void KernelFunction(…) {
  int tid = blockDim.x * blockIdx.x + threadIdx.x;
  int varA = aa[tid];
  int varB = bb[tid];
  C[tid] = varA + varB;
}
```

Slide credit: Hyesoon Kim

Vector Processor Disadvantages

-- Works (only) if parallelism is regular (data/SIMD parallelism)

++ Vector operations

-- Very inefficient if parallelism is irregular

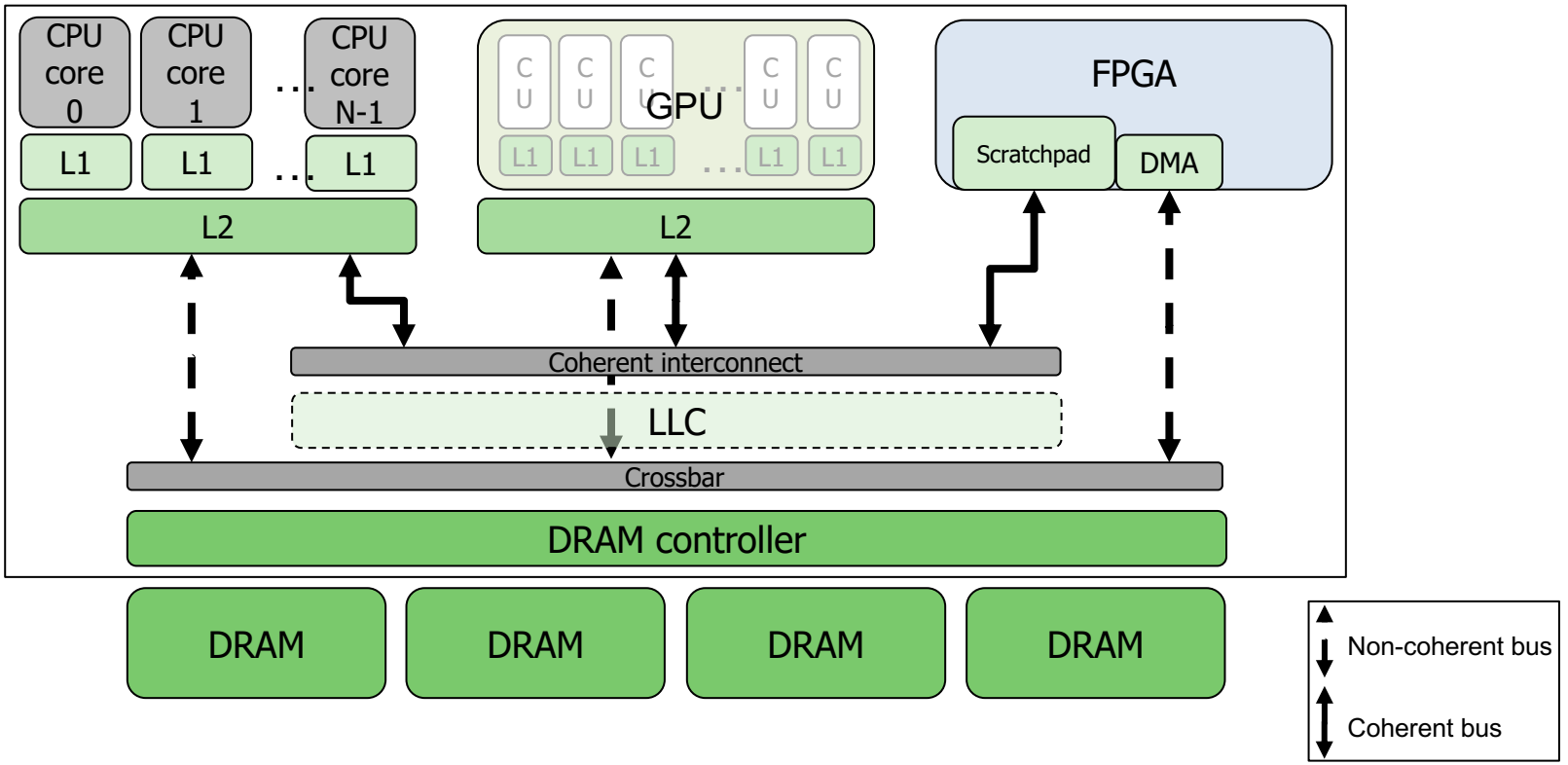
-- How about searching for a key in a linked list?

To program a vector machine, the compiler or hand coder must make the data structures in the code fit nearly exactly the regular structure built into the hardware. That's hard to do in first place, and just as hard to change. One tweak, and the low-level code has to be rewritten by a very smart and dedicated programmer who knows the hardware and often the subtleties of the application area. Often the rewriting is

Fisher, "Very Long Instruction Word architectures and the ELI-512," ISCA 1983.

Heterogeneous Computing Systems

- The end of Moore's law created the need for heterogeneous systems
  - More suitable devices for each type of workload
  - Increased performance and energy efficiency



Chai Benchmark Suite

- Heterogeneous execution on CPU, GPU, FPGA
- Collaboration patterns
  - 8 data partitioning benchmarks
  - 3 coarse-grain task partitioning benchmarks
  - 3 fine-grain task partitioning benchmarks
- Discrete (D) and Unified (U) versions
- Chai versions
  - CUDA and OpenCL for CPU+GPU
  - OpenCL for CPU+FPGA
  - CUDA-Sim for Gem5-GPU

https://chai-benchmarks.github.io

P&S Heterogeneous Systems

Programming Heterogeneous Computing Systems with GPUs and other Accelerators

Dr. Juan Gómez Luna
Prof. Onur Mutlu
ETH Zürich

Fall 2022

3 October 2022